

Estimating the Ice Thickness of Mountain Glaciers from Surface Topography and Mass-Balance Data

THÈSE N° 5940 (2013)

PRÉSENTÉE LE 27 SEPTEMBRE 2013
À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE D'ANALYSE ET DE SIMULATION NUMÉRIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Laurent MICHEL

acceptée sur proposition du jury:

Prof. K. Hess Bellwald, présidente du jury
Prof. M. Picasso, directeur de thèse
Prof. C. Ancey, rapporteur
Prof. H. Blatter, rapporteur
Prof. M. Funk, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

Quand vous avez un problème sérieux,
réfléchissez bien.
S'il y a une solution,
il n'y a pas lieu de s'inquiéter.
S'il n'y a pas de solution,
il est inutile de s'inquiéter.
— Dalai Lama

To my fiancée, my family, and my fiancée's parents. . .

Remerciements

Je remercie en premier lieu Jacques Rappaz qui m'a offert l'opportunité de travailler au sein d'un groupe très sympathique, sous la tutelle de Marco Picasso. Mon directeur de thèse a su se montrer disponible, à l'écoute, patient et de très bon conseil sur ce sujet particulièrement difficile. Les glaciologues de l'ETHZ, Martin Funk et Heinz Blatter, à l'origine du projet, m'ont permis de m'immerger dans un domaine qui m'était parfaitement inconnu, tant au niveau des connaissances mathématiques que physiques. Ils ont été d'une aide très précieuse et nos rencontres scientifiques à Zürich et Lausanne très productives. En outre, sans le travail de terrain de leurs collaborateurs, Daniel Farinotti et Andreas Bauder, et sans le travail informatique de tout le groupe de Marco Picasso et Jacques Rappaz, cette thèse n'aurait pas pu voir le jour. Enfin, je salue tout le travail de correction et de relecture des rapporteurs, qui a mené à des remarques très pertinentes autant sur le fond que la forme du contenu qui suit. Merci également à la présidente du jury pour sa disponibilité.

J'adresse encore quelques remerciements à mes collègues, Nicolas (Jacques Paul !) James, Stéphane Flotron, Thomas Hofer, Viljami Laurmaa, Jonathan Rochat et Diane Guignard, avec qui j'ai relativement peu travaillé mais néanmoins beaucoup rigolé, notamment lors de nos fameux "Computational Beer'n'Sandwich" à Sat. Merci également à Frédéric Besson, François Julmy et Florian Pittet qui m'ont bien diverti lors de nos petites soirées jeux.

Finalement, je remercie mes parents qui ont su me motiver à entreprendre ces études ainsi que ma (future) belle-famille pour les discussions ~~arrosées~~ glaciologiques. Tout particulièrement, je félicite Nathalie, ma fiancée, pour son soutien durant cette longue période de recherches.

Lausanne, le 06 Septembre 2013

LM

Abstract

The question addressed is the determination of a glacier's subglacial topography, given surface topography and mass-balance data. The input data can be obtained relatively easily for a large number of glaciers. Several methods essentially based on the shallow ice approximation are proposed, some of which are extended to Stokes ice flows. Two gradient-free, iterative methods are first introduced, namely the quasi-stationary inverse method, that relies on the apparent surface mass-balance description of glacier dynamics, and the transient inverse method, consisting in the iterative update of the bedrock topography proportionally to the surface topography misfit at the end of the glacier's considered evolution. Then, an optimal control algorithm is suggested that calculates the bedrock topography and some model parameters from surface observations through the minimization of a regularized misfit functional by means of a Lagrangian method. Numerical validations, along with sensitivity analyses and applications to real-world data are presented for each method.

Keywords: glacier, bedrock topography, shallow ice approximation, Stokes flow, shape optimization, optimal control, sensitivity analysis, shape derivative

Résumé

Le problème traité dans ce travail est le calcul du lit rocheux d'un glacier à partir de l'observation de sa topographie de surface et de son bilan de masse, qui sont des données relativement faciles à obtenir sur un grand échantillon de glaciers. Plusieurs méthodes sont proposées, essentiellement basées sur l'approximation de glace peu profonde. Certains algorithmes sont généralisés aux écoulements de Stokes. Deux méthodes itératives ne faisant pas usage d'un gradient sont d'abord décrites. La méthode inverse quasi-stationnaire repose sur le bilan de masse apparent d'un glacier. La méthode inverse évolutive consiste en l'ajustement itératif du lit rocheux proportionnellement à l'erreur faite sur la topographie de surface à la fin de l'évolution du glacier sur l'intervalle de temps considéré. Puis, un algorithme de contrôle optimal est développé. Il permet de calculer la position du lit rocheux et quelques paramètres du modèle, à partir d'observations sur la surface du glacier, par la minimisation d'une fonctionnelle de coût régularisée avec une méthode de Lagrangien. Ce rapport présente, pour chaque méthode, les validations numériques, analyses de sensibilité ainsi que des applications à des données mesurées sur des glaciers réels.

Mots-clés : glacier, lit rocheux, approximation de glace peu profonde, écoulement de Stokes, optimisation de forme, contrôle optimal, analyse de sensibilité, dérivée de forme

Contents

Remerciements	v
Abstract (English/Français)	vii
1 Introduction	1
2 Forward models	5
2.1 Problem definition	5
2.2 Ice flow equations	6
2.2.1 Stokes ice flow	7
2.2.2 Shallow ice flow	10
2.3 Free surface equation	11
2.3.1 Volume of fluid formulation	12
2.3.2 Shallow ice formulation	13
2.4 Surface mass-balance	14
2.5 Conclusion	15
3 Forward numerical models	17
3.1 Methodology	17
3.1.1 Stokes ice flow	17
3.1.2 Shallow ice flow	18
3.2 Stokes ice flow	18
3.2.1 Ice velocity computation	18
3.2.2 Ice transport	20
3.3 Shallow ice flow	21
3.4 Validation	24
3.5 Conclusion	30
4 Sensitivity analysis	31
4.1 Changing parameters on a fixed geometry	32
4.2 Influence of subglacial bumps on various geometries	35
4.3 Conclusion	40

5	Transient Inverse Method	43
5.1	Parallel-sided slab estimation	44
5.2	Stationary Inverse Method	46
5.3	Quasi-stationary Inverse Method	49
5.3.1	Two-dimensional formulation	49
5.3.2	Three-dimensional formulation	50
5.4	Transient Inverse Method	51
5.5	Numerical results	53
5.5.1	Stationary Inverse Method	53
5.5.2	Quasi-stationary Inverse Method	56
5.5.3	Transient Inverse Method	61
5.6	Sensitivity analysis	69
5.6.1	Two-dimensional shallow ice flows	69
5.6.2	Three-dimensional Stokes ice flows	70
5.7	Application to real-world data	72
5.7.1	Quasi-stationary Inverse Method	72
5.7.2	Transient Inverse Method	73
5.8	Conclusion	76
6	Optimal control of the shallow ice model	79
6.1	Two-dimensional description	80
6.2	Three-dimensional formulation	85
6.2.1	Basic algorithm	85
6.2.2	Computation of model parameters from surface topography data	91
6.2.3	Incorporation of velocity data	93
6.2.4	Automatic differentiation	95
6.3	Numerical results	97
6.3.1	Two-dimensional formulation	97
6.3.2	Three-dimensional formulation	102
6.4	Sensitivity analysis	110
6.4.1	Two-dimensional algorithm	110
6.4.2	Three-dimensional algorithm	112
6.5	Application to real-world data	116
6.6	Perspectives	117
6.7	Conclusion	119
7	Conclusion	121
A	Linear systems of the forward models	123
A.1	Stokes ice flow	123
A.2	Shallow ice flow	125
A.2.1	Two-dimensional schemes	125
A.2.2	Three-dimensional schemes	126
B	Linear systems of the dual problems	129
B.1	Two-dimensional formulation	129
B.1.1	Semi-implicit time discretization	129
B.1.2	Implicit time discretization	129
B.2	Three-dimensional formulation	130

B.2.1	Semi-implicit time discretization	130
B.2.2	Implicit time discretization	130
Bibliography		139
Curriculum Vitae		141

1 Introduction

A few centuries ago, people living in the mountains used to fear the glaciers surrounding them as they caused repeated avalanches and floods from glacial lakes. For example, due to its extent at that time, Aletsch glacier, Swiss Alps, used to threaten many homes, so that Pope Innocent XI was asked in 1678 by the inhabitants of its neighbouring hamlets to make it retreat by the hands of God, for the sake of human lives.

The Pope may have been heard, as Aletsch glacier is still retreating today. Most of the alpine glaciers have globally lost a large amount of ice mass, especially since the 1850's, as far as available pictures and tracks of ancient moraines can testify. For example, at that time, Rhône glacier, Swiss Alps, had its tongue terminus located beside the famous Hotel Glacier du Rhône, Gletsch VS, which used to be filled with many tourists fascinated by the glacier. Figure 1.1 shows how far the glacier's tongue extended in 1900 into Gletsch village and how much it retreated from 1900 to 1920. The red arrows evince the tongue's position in 1850, showing how critical the retreat was already one century ago. Since then, Rhône glacier has never stopped retreating and it is now confined further up Furkastrasse, above Hôtel Belvédère, near which a small lake has recently formed.



Year 1900



Year 1920

Figure 1.1 – Available photographs of Rhône glacier in 1900 and 1920 [84, 30]. At the bottom of the pictures, this is Gletsch village with the Hotel Glacier du Rhône on the left. The red arrows show the glacier's position in 1850.

In 2009, the citizens of Fiesch and Fieschertal contacted Pope Benedict to reverse the vow of 1678, as they noticed that the retreating glacier has become dangerous for human beings. Global warming, which seems correlated to the beginning of the industrial revolution, is accelerating the glaciers' retreat. With the increase of the Earth's global

temperature, glaciers are melting, with non-negligible environmental consequences on local and global scales [20, 32]. Locally, the Swiss tourism, agriculture, and hydro-power industry already measure the economical impacts of the phenomenon. For example, the construction of new dams must be envisaged because of the recently formed lakes, in order to prevent the neighbouring inhabitations from being flooded. On a global scale, the wildlife and very low-lying lands are affected by the sea-level rise due to the melting of large polar ice sheets.

For these reasons, popular topics in glaciology include climatic and hazard forecasts, as well as predictions of the formation of new lakes in subglacial depressions. They all require the accurate modeling of glacier dynamics that depends on the precision of the boundary conditions and model parameters. For instance, initializing a glacier model with a rough approximation of its bedrock topography may result in possibly significant errors in the sought predictions. Moreover, the glacier models' building blocks, namely the ice rheology, the surface mass-balance, and the sliding law, depend on parameters whose physical values can hardly be measured accurately and are therefore still the subject of current research [49, 61, 5, 62, 4, 68, 53, 94, 16].

Once the boundary conditions and the model parameters are known, the numerical simulation of glaciers can provide the desired forecasts. The most complete description of ice dynamics is provided by a non-linear Stokes approximation of ice flow coupled with an evolution equation for the glacier's surface that transports ice following the Stokes velocity [49, 66]. Past reconstruction and future predictions on some Swiss glaciers were successfully performed with this three-dimensional model [67, 68]. However, this complete description relies on large computer resources and makes model inversions difficult. Hence, many lower order models of ice flow have been developed in the past decades, among which the widely-spread shallow ice and the first order approximations.

The volume of mountain glaciers is an important component for modeling studies of future glacier changes and their impacts on the alpine environment (*e.g.* [96]). While the total glacier volume is often estimated without knowledge of the subglacial topography through the so-called “volume-area scaling” (*e.g.* [23, 6]), this information is indispensable for many glaciological studies. A glacier's bed topography can be determined with geophysical methods, but, for practical reasons, measurements can only be performed at selected locations and for a limited sample of glaciers [25]. Therefore, several theoretical approaches to determine the subglacial topography have been proposed. The assumptions of a parabolic cross section and perfectly plastic behaviour of glacier ice supplied a first estimation [91], where the shear stress at the glaciers' base is supposed constant and close to the plastic yield stress of ice. A similar approach was suggested, based on the parallel-sided slab approximation (*e.g.* [48], Chapter 7.2), in which the glacier-specific plastic yield stress is empirically inferred as a function of the glacier's elevation range and then related to the glacier's thickness and slope [54, 93]. Recently, an extended perfect plasticity method was presented that accounts for the effect of side drag on the stress balance [75]. An alternative method relying on mass turnover and parallel-sided slabs was also proposed [34]. The subglacial topography was also estimated by means of inverse approaches, in which indirect surface velocity measurements were used (*e.g.* [50, 111, 98, 99, 85, 33, 83]). A completely different procedure to estimate the subglacial topography was designed on artificial neural networks [27].

On subglacial topography estimation by means of optimal control, the oldest contribution is certainly the interpolation method that seeks the best solution fitting the measured data, under the constraint that it also minimizes potential and curvature [63]. Subsequent work on optimal control algorithms that infer the ice thickness distribution of a glacier is rather sparse but literature has lately become more abundant on those that minimize the surface velocity misfit on steady geometries [85, 83]. Without direct consideration of surface velocity data, the most recent proposition consists in the minimization of the discrepancy between the actual ice thickness and the quasi-stationary, shallow ice corresponding value for a three-dimensional geometry [26]. Optimal control is more often used to determine basal sliding law and rheology coefficients [116, 74, 70, 82, 71, 5, 4, 94, 53, 81], where the misfit between the computed and measured surface velocities is minimized with the help of two and three-dimensional flow models of arbitrary order. Most of the aforementioned methods aiming at the estimation

of the bedrock topography elevation make use of surface velocity data, which can lead to an underdetermined problem when it comes to the computation of both the basal topography and model parameters like rheology, surface mass-balance, and sliding coefficients. Moreover, basically all the existing optimal control methods that solve this problem deal with stationary or quasi-stationary geometries exclusively, except the contributions [46, 57] that make use of both a primal–dual and an automatic differentiation algorithm.

Three main results are presented in this thesis: First, a sensitivity analysis of both the Stokes and shallow glacier forward models to all the parameters and the bedrock topography is reported for several alpine glaciers, where both ice flow models are compared. The ill-posedness of the inverse problem of computing a glacier's subglacial topography from surface observations is clearly evinced with various numerical experiments. Second, two gradient-free inverse methods to infer the bedrock topography of a glacier from its transient evolution are developed. The quasi-stationary inverse method is based on the apparent surface mass-balance description of the glacier's time evolution and hence extends the well-working procedure exposed in [34] to a more automatic algorithm that requires less pre-processing work with numerical schemes that are well-known in glaciology. Given the digital elevation model (DEM) of the surface topography together with the surface mass-balance and the sliding law parameters, it reduces to the solving of a regularized hyperbolic equation. In turn, the transient inverse method aims at taking the time dependence of a glacier's evolution more accurately. Indeed, the apparent surface mass-balance mentioned in the previous method's description supposes that the evolution of the ice thickness is uniform in time, which is a crude approximation. The central idea of the transient inverse method is the iterative update of the bedrock topography with the surface topography misfit after a transient evolution. The algorithm was already applied to two-dimensional, stationary, Stokes flows [58] and to three-dimensional, transient, shallow ice sheets [114]. Last, an optimal control algorithm to compute a glacier's subglacial topography and some model parameters from surface observations is formulated for shallow glaciers. It consists in the minimization of a discrete regularized misfit functional by means of a primal–dual method and constitutes the most elaborated method developed in this thesis. It is flexible, versatile, and fast, at the expense of requiring more hand coding and a better initial guess. Moreover, the procedure can be extended to higher order models.

Furthermore, a few minor novelties are introduced in both the forward and inverse approaches in glaciology. In the context of forward simulations, a new numerical scheme is introduced that is more stable than its widely spread, semi-implicit-in-time version, for the simulation of three-dimensional, shallow glaciers. Moreover, a sliding law is proposed that models in the simplest possible way a dependence of the sliding in the bedrock topography elevation. This choice is motivated by the fact that sliding is induced by melted water underneath the glacier and that the quantity of such water decreases with altitude. In this model, the sliding law depends on the so-called sliding altitude and the sliding coefficient. All the points of the bedrock topography that lie below this altitude are defined to be sliding points, contrary to those located above it. Such a description is useful to perform a sensitivity analysis of the models investigated in this thesis and to show the limitations of the derived inverse approaches. Moreover, it provides a two-parameter modeling of sliding that may be useful when only little data are available as input to the inversion algorithms. This form of the sliding law is however purely theoretical and does not rely on concrete physical experiments. Besides, the three-parameter surface mass-balance function, consisting in a melting and an accumulation rate below and above the equilibrium line altitude respectively, is extended to a smooth function, for the sake of derivation in the proposed inverse approaches based on optimal control techniques.

The numerical experiments performed throughout this thesis show that the quasi-stationary inverse method provides a reasonable estimation of the subglacial topography of a glacier with very little work and computer resources. The transient inverse method is reliable for shallow ice flows but not any more for Stokes flows. The best algorithm is the shape optimization method. However, it is formulated here for shallow ice flows only.

The thesis' outline is the following: The equations governing the Stokes and shallow ice flows are first introduced. Besides, the volume of fluid formulation of the ice thickness equation is recalled which allows a more local, three-

dimensional mass conservation. Next, the traditional numerical methods that solve these problems are presented, namely a finite element formulation for the Stokes problem, a time-splitting scheme for the volume of fluid equation, and finite difference schemes for its shallow ice approximation. The volume of fluid equation is splitted into a transport equation and a surface mass-balance equation. The former is solved with a forward characteristic method and complemented with algorithms reducing the undesirable effects of numerical diffusion and compression, while the latter is solved with a simple algorithm that stacks the volume of fluid function. Chapter 4 exposes a sensitivity analysis of the forward models. Several numerical experiments are investigated, such as the simulation of a glacier's evolution with different model parameters. Moreover, the influence of a bump in the bedrock topography is studied on several glacier geometries inspired from real-world data. Next, Chapter 5 introduces the quasi-stationary and transient inverse methods. In Chapter 6, the shape optimization algorithm is formulated, also in two and three space dimensions. Although it is based on the shallow ice equations, an application of the latter method to Stokes flows is proposed. The methods in Chapters 5 and 6 are first developed in two space dimensions and then extended to three dimensions. Finally, general conclusions are drawn.

2 Forward models

The time evolution of a glacier is described by a free surface equation where the ice – air interface, namely the surface topography, has to be determined. It basically consists in ice transport where the ice velocity obeys the stationary, non-linear Stokes equations which constitute the current best approximation of ice dynamics [49, 119]. Moreover, among the algorithms that allow for the tracking of the ice volume’s interface, the Eulerian volume of fluid method is advocated, where the characteristic function of ice is transported following the Stokes ice velocity. This model successfully allowed for the past reconstruction and future predictions on Rhône and Aletsch glaciers [67, 68]. However, solving the coupled volume of fluid and Stokes equations is often time-consuming and memory-demanding, especially in inverse approaches. Therefore, approximations of the non-linear Stokes problem are here considered that can reproduce the model’s most important features, as for example the shallow ice approximation that is suited to thin glaciers.

Many textbooks exist on both the free surface and Stokes problems [97, 80, 60, 71, 48, 92]. This chapter only provides a short summary of the equations arising in glacier dynamics and relevant for the optimization algorithms presented in the next chapters. First, the stationary, non-linear Stokes equations and their approximations are recalled, among which the shallow ice approximation, that describe ice flow. Then, the ice thickness equation and its volume of fluid correspondent are introduced. They are the equations that make the free surface evolve under the influence of climatic input and ice motion.

2.1 Problem definition

A three-dimensional, time-dependent glacier ice volume $\Omega(t)$ enclosed in a cavity $\Lambda = \Omega_{\perp} \times [\underline{Z}, \overline{Z}]$, with $\Omega_{\perp} = [0, L_x] \times [0, L_y]$ (see Figure 2.1) is considered. In what follows, the time-independent domain Ω_{\perp} is termed “glacier map domain”. The ice volume is bounded by the glacier’s bedrock and surface topographies b and s respectively, which are both functions of the horizontal coordinates x and y . Glacier isostasy (*e.g.* [48], Chapter 8) is neglected, hence only the surface topography actually changes with time. Accordingly, the ice thickness is defined by $\mathcal{H} = s - b$. At initial time t_i , the surface topography is known and denoted by s_i . All the equations in this thesis are expressed in the Cartesian system of axes (x, y, z) .

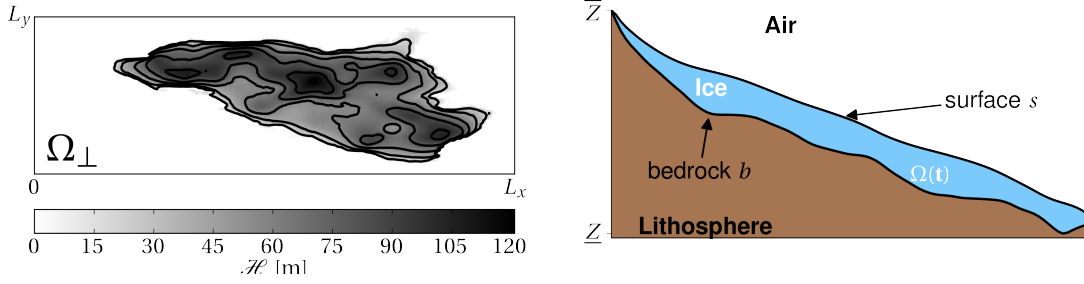


Figure 2.1 – Left: Ice extent in the (x, y) -plane of Silvretta glacier, Swiss Alps. The gray levels represent the ice thickness distribution. The map domain Ω_{\perp} consists in the whole rectangle $[0, L_x] \times [0, L_y]$. Right: glacier profile along a flow line. The bedrock topography b is the ice – lithosphere interface, while the surface topography is the ice – air interface. The ice domain is depicted in blue.

The glacier's dynamics are governed by the ice mass and momentum conservation. Moreover, it is influenced by the ice rheology and the glacier's surrounding climate. The former is characterized by Glen's flow law [44]. The climatic information is encoded in the so-called surface mass-balance function \mathcal{B} , a representation of which is given on Figure 2.2. Basically, two regions of the glacier's surface are distinguished where ice accumulates or melts, delimited by the so-called equilibrium line altitude (ELA). Ice flow equations describe the ice dynamics while an ice thickness equation deals with the glacier's evolution under the climatic conditions it is subjected to. The forward problem is therefore the following: *Given a bedrock b and a surface topography s_i at initial time t_i , and given the ice rheology as well as the surface mass-balance \mathcal{B} , find the ice surface topography s_f at final time t_f .* The governing ice flow and free-surface equations are explicated in the next sections. The surface mass-balance function used throughout this thesis is expressed in the end of the chapter.

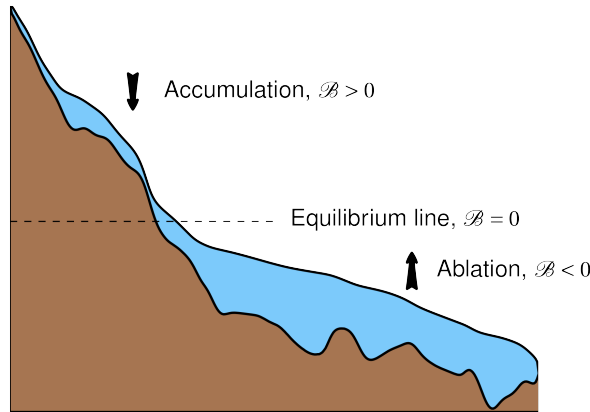


Figure 2.2 – Illustration of the surface mass-balance resulting from the climate the glacier is subjected to, along a flow line of Rhône glacier, Swiss Alps.

2.2 Ice flow equations

Ice is a non-Newtonian, incompressible fluid of extremely large viscosity compared to most of all-day fluids. As illustrated by Table 2.1, it is 10^{16} times more viscous than temperate water. Such a high viscosity makes ice move very slowly, of the order of magnitude of 100 meters per year (m a^{-1} in this thesis) in Swiss Alps glaciers. In a glacier,

the shear stress τ is linked to the shear rate $\dot{\gamma}$ through the relation (*e.g.* [48], Chapter 4)

$$\tau = \eta \dot{\gamma}, \quad (2.1)$$

where η is the shear viscosity. Ice dynamics is described by the stationary, non-linear Stokes diffusion equations [49, 119], where the viscosity is ruled by Glen's non-linear law in the ice velocity [44]. Various models are proposed for the estimation of the latter velocity. First, the Stokes ice flow equations are introduced, whose analytical solutions cannot be given explicitly in general, hence the finite-element method presented in the next chapter. Second, the Stokes ice model is simplified to a shallow ice model, where the ice velocity equations can be solved explicitly. In the next section, the free surface equation uses it to transport the ice, which makes the glacier's surface evolve in time.

Table 2.1 – Typical orders of magnitude of the viscosity (in Pa · s) of some all-day life fluids.

<i>Air</i>	10^{-5}
<i>Water</i>	10^{-3}
<i>Honey</i>	10
<i>Ice</i>	10^{13}

2.2.1 Stokes ice flow

Ice is considered as an incompressible, non-Newtonian fluid whose rheology is described by the symmetric, Cauchy stress tensor

$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix} = -p \text{id} + 2\eta D \quad (2.2)$$

where p is the ice pressure, id the identity matrix, D the strain-rate matrix

$$D = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad (2.3)$$

\mathbf{u} the ice velocity, and η the ice (shear) viscosity. It satisfies Glen's flow law [44]

$$2^{\frac{n+1}{2}} A |D|^{n-1} \eta^n + 2 A \sigma_0^{n-1} \eta - 1 = 0, \quad (2.4)$$

where $n \geq 1$ is Glen's exponent, A the rate factor, and σ_0 is a small positive constant that prevents the viscosity from going to the infinity when the quantity

$$|D| = \sqrt{D : D} = \sqrt{\text{tr}(D^2)} \quad (2.5)$$

approaches zero. For alpine glaciers, these parameters have orders of magnitude $n = 3$, $A = 0.08 \text{ bar}^{-3} \text{ a}^{-1}$, and $\sigma_0 = \sqrt{0.1} \text{ bar}$.

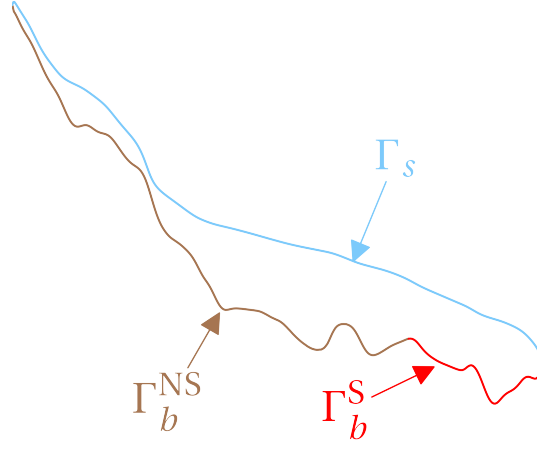


Figure 2.3 – Glacier's boundaries along a flow line of Rhône glacier, Swiss Alps. Its ice – air interface (2.6) is denoted by Γ_s while its ice – lithosphere interface, the bedrock (2.7), by Γ_b . The latter is decomposed into a sliding and a sticky part. On the sliding part Γ_b^S , only the tangential components of the ice velocity are non-zero. On the sticky part Γ_b^{NS} , the ice velocity is identically zero.

The boundary conditions are the following: Let us assume that no force acts on the top surface, namely the ice – air interface (see Figure 2.3)

$$\Gamma_s(t) = \{(x, y, z) \in \Omega(t) | z = s(x, y, t)\}. \quad (2.6)$$

Moreover, the ice – lithosphere interface, namely the bedrock,

$$\Gamma_b = \{(x, y, z) \in \Omega(t) | z = b(x, y)\} \quad (2.7)$$

is the reunion of Γ_b^{NS} and Γ_b^S where the glacier is assumed to stick to and slide on the subglacial rock respectively.

Due to the typical length scales arising in a glacier, the time-dependent, non-linear Navier-Stokes equations reduce to the stationary, non-linear Stokes equations. Indeed, since a glacier has a typical horizontal extension of 10^3 km, a vertical extension of 1 km, a horizontal velocity of $100 \text{ m} \cdot \text{a}^{-1}$, a vertical velocity of $0.1 \text{ m} \cdot \text{a}^{-1}$, and is observed over a time scale of 10^4 a, the inertial and acceleration terms are small compared to the gradient-of-pressure term. With the boundary conditions, the ice dynamics equations are cast into

$$\left\{ \begin{array}{ll} -2\nabla \cdot (\eta D) + \nabla p = \rho \mathbf{g}, & \text{in } \Omega(t) \\ \nabla \cdot \mathbf{u} = 0, & \text{in } \Omega(t) \\ -p\mathbf{n} + 2\eta D \cdot \mathbf{n} = 0, & \text{on } \Gamma_s(t) \\ \mathbf{u} = 0, & \text{on } \Gamma_b^{NS} \\ \mathbf{u} \cdot \mathbf{t}_i = -2\eta \frac{(D \cdot \mathbf{n}) \cdot \mathbf{t}_i}{\alpha}, & \text{on } \Gamma_b^S, \quad i = 1, 2, \\ \mathbf{u} \cdot \mathbf{n} = 0, & \text{on } \Gamma_b^S, \end{array} \right. \quad (2.8)$$

where \mathbf{t}_i , $i = 1, 2$, are the tangential vectors to the ice volume, \mathbf{n} its normal vector, η the ice viscosity (2.4), and

$$\alpha = \frac{C}{(\|\mathbf{u}\| + s_0)^{1-\frac{1}{n}}} \quad (2.9)$$

defines Weertman's sliding law [117, 67]. In (2.9), s_0 is a regularization parameter and C some position-dependent

function. Equation (2.8)₁ describes the conservation of momentum, (2.8)₂ the conservation of mass, and the remaining expressions are the boundary conditions.

Many ways of expressing the sliding law exist [65, 36, 39, 104, 115, 105]. Instead of assuming C to be a piecewise constant [65, 67], sliding is here supposed to be induced by melted water flowing below the ice mass and the amount of such water actually decrease with altitude [92, 106], mainly because temperature diminishes. Such a description is not traditional in glaciology and is introduced here only to provide a simplified sliding model whose amount of parameters is small enough to make it useful in an optimal control problem where only little data are available. This involves the introduction of the so-called “sliding altitude” z_{sl} (see Figure 2.4). Sliding occurs only at the locations where the bedrock elevation is below the sliding altitude. In addition, for the sake of simplicity, let us assume that the effects of sliding on ice velocity increase linearly with the distance from the sliding altitude at order zero. This specializes the sliding coefficient C to

$$\frac{1}{C} = ([z_{sl} - b]^+)^{\frac{1}{n}} \mathcal{C}, \quad (2.10)$$

where \mathcal{C} is a constant, $[z_{sl} - b]^+$ is the positive part of $z_{sl} - b$, *i.e.*

$$[z_{sl} - b]^+ = \begin{cases} z_{sl} - b, & \text{if } z_{sl} > b \\ 0, & \text{otherwise.} \end{cases} \quad (2.11)$$

The shallow ice expressions (2.13)–(2.16) clarify this motivation behind function C that makes the ice velocity linearly dependent on $[z_{sl} - b]^+$.

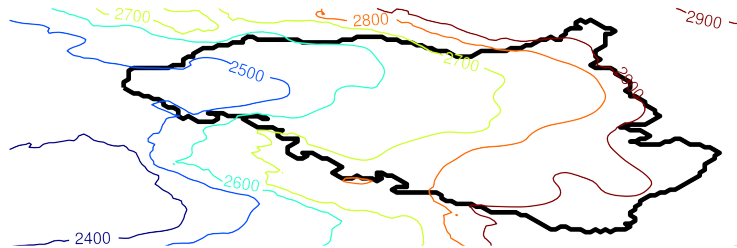


Figure 2.4 – Contour lines of the bedrock topography of Silvretta glacier. They delineate possible sliding altitudes z_{sl} below which sliding is assumed.

2.2.2 Shallow ice flow

Because solving the Stokes equations (2.8) is time-consuming, small contributions to the ice velocity and pressure may be neglected to reduce their complexity. Table 2.2 lists the different approximations with the corresponding components of tensor (2.2) [60, 48]. In the hydrostatic approximation (HSA), the shear stresses σ_{xz} and σ_{yz} are supposed small compared to σ_{zz} . Therefore, due to condition (2.8)₃, the vertical normal stress σ_{zz} is hydrostatic and the pressure is eliminated:

$$p = \rho g(s - z) - 2\eta \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \right). \quad (2.12)$$

Table 2.2 – Traditional approximations of the Cauchy stress tensor σ components (2.2) in the Stokes equations (2.8): hydrostatic approximation (HSA), first order approximation (FOA), shallow ice approximation (SIA), and first order plain strain approximation (FOPSA).

$\sigma_{xx} =$	$-p + 2\eta \frac{\partial u_x}{\partial x}$	$\sigma_{xz} = \eta \left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \right)$	(HSA)
$\sigma_{yy} =$	$-p + 2\eta \frac{\partial u_y}{\partial y}$	$\sigma_{yz} = \eta \left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \right)$	
$\sigma_{zz} =$	$-\rho g(s - z)$	$\sigma_{xy} = \eta \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right)$	
$\sigma_{xx} =$	$-p + 2\eta \frac{\partial u_x}{\partial x}$	$\sigma_{xz} = \eta \frac{\partial u_x}{\partial z}$	(FOA)
$\sigma_{yy} =$	$-p + 2\eta \frac{\partial u_y}{\partial y}$	$\sigma_{yz} = \eta \frac{\partial u_y}{\partial z}$	
$\sigma_{zz} =$	$-\rho g(s - z)$	$\sigma_{xy} = \eta \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right)$	
$\sigma_{xx} =$	$-p$	$\sigma_{xz} = \eta \frac{\partial u_x}{\partial z}$	(SIA)
$\sigma_{yy} =$	$-p$	$\sigma_{yz} = \eta \frac{\partial u_y}{\partial z}$	
$\sigma_{zz} =$	$-\rho g(s - z)$	$\sigma_{xy} = 0$	
$\sigma_{xx} =$	$-p + 2\eta \frac{\partial u_x}{\partial x}$	$\sigma_{xz} = \eta \frac{\partial u_x}{\partial z}$	(FOPSA)
$\sigma_{yy} =$	$-p$	$\sigma_{yz} = 0$	
$\sigma_{zz} =$	$-\rho g(s - z)$	$\sigma_{xy} = 0$	

At this point, many further approximations can be done. For instance, because of its typical order of magnitude, $\frac{\partial u_z}{\partial x}$ is small compared to $\frac{\partial u_x}{\partial z}$, as is $\frac{\partial u_z}{\partial y}$ compared to $\frac{\partial u_y}{\partial z}$. Hence the components of the Cauchy tensor in the HSA can be simplified to the first order approximation (FOA) [10, 28, 71]. The advantage of these simplifications is that now the component u_z of the velocity is fully decoupled from the other ones, for its derivatives are not present in the Cauchy tensor σ any more.

The flow regime can further be assumed to be essentially a simple, bed-parallel shear. This is the shallow ice approximation (SIA) which is the essential approximation used throughout this thesis. In this approximation, assume the sliding is not regularized ($s_0 = 0$ in (2.9)). The particularity of SIA is that the ice velocity components can be expressed as the following functions of the ice thickness \mathcal{H} and the glacier's surface topography $s = b + \mathcal{H}$ [48, Chapter 5.4]:

$$u_x(x, y, z) = - \left(\Gamma_s(x, y) \mathcal{H}^n(x, y) + \frac{n+2}{n+1} \Gamma \left(\mathcal{H}^{n+1}(x, y) - (s(x, y) - z)^{n+1} \right) \right) \left\| \nabla s(x, y) \right\|^{n-1} \frac{\partial s}{\partial x}(x, y) \quad (2.13)$$

$$u_y(x, y, z) = - \left(\Gamma_s(x, y) \mathcal{H}^n(x, y) + \frac{n+2}{n+1} \Gamma \left(\mathcal{H}^{n+1}(x, y) - (s(x, y) - z)^{n+1} \right) \right) \left\| \nabla s(x, y) \right\|^{n-1} \frac{\partial s}{\partial y}(x, y) \quad (2.14)$$

$$u_z(x, y, z) = - \int_b^z \left(\frac{\partial u_x}{\partial x}(x, y, \bar{z}) + \frac{\partial u_y}{\partial y}(x, y, \bar{z}) \right) d\bar{z}, \quad (2.15)$$

where

$$\Gamma = 2 \frac{A(\rho g)^n}{n+2} \quad \text{and} \quad \Gamma_s = \left(\mathcal{C} \rho g \right)^n [z_{sl} - b]^+ \quad (2.16)$$

are diffusion coefficients and $s = b + \mathcal{H}$. Of particular interest in this thesis are the surface ice velocity components

$$u_x|_{z=s} = - \left(\Gamma_s + \frac{n+2}{n+1} \Gamma \mathcal{H} \right) \mathcal{H}^n \left\| \nabla s \right\|^{n-1} \frac{\partial s}{\partial x} \quad (2.17)$$

$$u_y|_{z=s} = - \left(\Gamma_s + \frac{n+2}{n+1} \Gamma \mathcal{H} \right) \mathcal{H}^n \left\| \nabla s \right\|^{n-1} \frac{\partial s}{\partial y} \quad (2.18)$$

$$u_z|_{z=s} = \nabla \cdot \left(\mathcal{D} \nabla s \right) + \mathbf{u}|_{z=s} \cdot \nabla s, \quad (2.19)$$

where the total diffusivity \mathcal{D} consists in two contributions, \mathcal{D}_f and \mathcal{D}_s , due to flow dynamics and the sliding law respectively, namely

$$\mathcal{D} = \mathcal{D}_f + \mathcal{D}_s, \quad (2.20)$$

where

$$\mathcal{D}_f = \Gamma \mathcal{H}^{n+2} \left\| \nabla(b + \mathcal{H}) \right\|^{n-1} \quad \text{and} \quad \mathcal{D}_s = \Gamma_s \mathcal{H}^{n+1} \left\| \nabla(b + \mathcal{H}) \right\|^{n-1}. \quad (2.21)$$

The vertical velocity component u_z is deduced from the horizontal components u_x , u_y and the assumption of negligible isostasy and basal melting. Most of the existing Swiss glacier ice velocity measurements can be compared to the horizontal surface ice velocity \mathcal{U}

$$\mathcal{U} = \sqrt{u_x|_{z=s}^2 + u_y|_{z=s}^2} = \left(\frac{n+2}{n+1} \Gamma \mathcal{H} + \Gamma_s \right) \mathcal{H}^n \left\| \nabla s \right\|^n. \quad (2.22)$$

Ice flow can be approximated further. For glaciers that are much longer than their average width, the complex, three-dimensional problem can be reduced to the simpler, two-dimensional problem of solving for the fields along the glacier's principal flow line. In this case, the coordinate axes (x, z) follow the flow line, and no variation transversal to it is considered. In this first order plane strain approximation, the Cauchy stress tensor components are given by (FOPSA) in Table 2.2.

Finally, let us mention the parallel-sided slab approximation (PSSA) [48, 34]. As the name indicates, this approximation consists in taking the glacier as an inclined parallelepiped. This is a particular case of the two-dimensional SIA where the ice thickness is kept constant everywhere in the domain. Such an approximation is used to roughly estimate the ice thickness in the central part of small valley glaciers. Sometimes, the glaciologists split the glaciers into ice catchments where the PSSA approximation is locally applied [34].

2.3 Free surface equation

The free surface equation is basically an equation that transports ice following its velocity computed in the previous section. Because of mass conservation, the glacier's proper dynamics and surrounding climate make the ice

thickness evolve according to the so-called ice thickness equation

$$\left\{ \begin{array}{lll} \frac{\partial \mathcal{H}}{\partial t} + \nabla \cdot \mathbf{Q} = \mathcal{B}, & \text{in } \Omega_{\perp} & \times [t_i, t_f] \\ \mathcal{H} \geq 0, & \text{in } \Omega_{\perp} & \times [t_i, t_f] \\ \mathcal{H} = 0, & \text{on } \partial\Omega_{\perp} & \times [t_i, t_f] \\ \mathcal{H} = s_i - b, & \text{in } \Omega_{\perp} & \times \{t_i\}, \end{array} \right. \quad (2.23)$$

where \mathcal{B} is applied only where $\mathcal{H} > 0$, \mathbf{Q} is the volume flux whose components are

$$Q_x(x, y) = \int_b^s u_x(x, y, z) dz \quad (2.24)$$

$$Q_y(x, y) = \int_b^s u_y(x, y, z) dz, \quad (2.25)$$

and \mathbf{u} is the ice velocity whose components either satisfy the Stokes equations (2.8) or are the analytical, shallow ice expression (2.13)–(2.15). When a Stokes velocity is available, the ice thickness equation (2.23) is reformulated as a volume of fluid (VOF) equation, because this improves mass conservation. In the case of a shallow ice flow, equation (2.23) is solved as it is.

2.3.1 Volume of fluid formulation

Equation (2.23)₁ ensures mass conservation in the glacier's ice columns. A more local mass conservation can be achieved with the introduction of the volume of fluid (VOF) function that is the characteristic function of the ice volume $\Omega(t)$:

$$\varphi(x, y, z, t) = \begin{cases} 1, & \text{if } (x, y, z) \in \Omega(t) \\ 0, & \text{otherwise.} \end{cases} \quad (2.26)$$

One can show that if φ is the solution of the VOF equation

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = \mathcal{B} \delta_{\Gamma_s}, \quad \text{in } \Lambda \times [t_i, t_f], \quad (2.27)$$

where δ_{Γ_s} is the surface density defined by

$$\int_V f \delta_{\Gamma_s} dV = \int_{\Gamma_s \cap V} f dS \quad (2.28)$$

for any volume V and continuous function f , then the ice thickness

$$\mathcal{H}(x, y) = \int_{\underline{Z}}^{\overline{Z}} \varphi(x, y, z) dz \quad (2.29)$$

satisfies (2.23)₁ [69, 66]. Equation (2.27) improves the physics of the problem, since it can describe topology changes such as ice cliffs and overhanging ice or simply the situation depicted by Figure 2.5. In this example, the glacier's initial contour, on the left, evolves towards the contour on the right. Ice melted so much during the evolution that the original ice mass splitted into two separate masses.

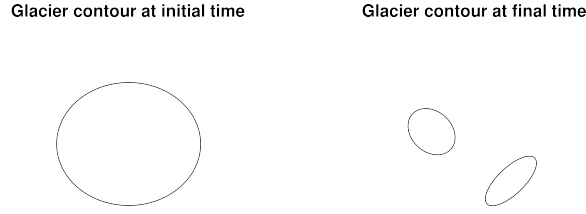


Figure 2.5 – Example change of topology that can be handled easily with the VOF equation (2.27). The glacier's contour evolves from the one represented on the left at initial time to that on the right at final time.

2.3.2 Shallow ice formulation

Now, since the ice velocity components u_x, u_y, u_z can be expressed as the functions (2.13), (2.14), and (2.15) of the ice thickness \mathcal{H} , the ice volume flux (2.24)–(2.25) becomes

$$\mathbf{Q} = -\mathcal{D}\nabla s, \quad (2.30)$$

turning the ice thickness equation (2.23) to the three-dimensional shallow ice equation for the ice thickness

$$\left\{ \begin{array}{lll} \frac{\partial \mathcal{H}}{\partial t} = \nabla \cdot (\mathcal{D}\nabla(b + \mathcal{H})) + \mathcal{B}, & \text{in } \Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} \geq 0, & \text{in } \Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} = 0, & \text{on } \partial\Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} = s_i - b, & \text{in } \Omega_{\perp} \times \{t_i\}. \end{array} \right. \quad (2.31)$$

This equation provides results in three space dimensions although it is solved in the (x, y) -plane only. In what follows, (2.31) is referred to as three-dimensional shallow ice (SIA3D) equation.

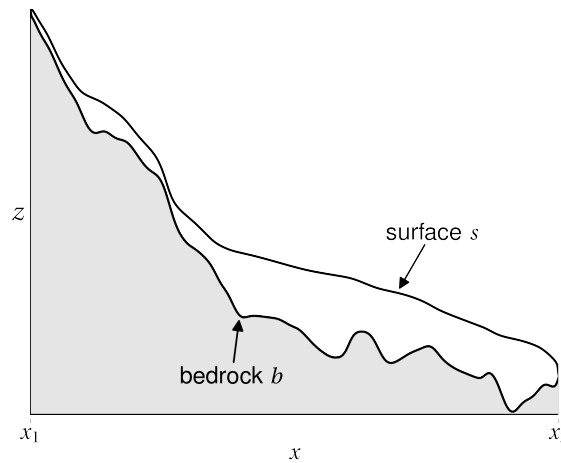


Figure 2.6 – Two-dimensional glacier in the Cartesian system of axes (x, z) .

Wide use is done of the two-dimensional shallow ice approximation (SIA2D) throughout this thesis to allow for the formal understanding of the more complex three-dimensional algorithms. The corresponding equation is the

two-dimensional specialization of (2.31) (see notations on Figure 2.6):

$$\left\{ \begin{array}{lll} \frac{\partial \mathcal{H}}{\partial t} = \frac{\partial}{\partial x} \left(\mathcal{D} \frac{\partial}{\partial x} (b + \mathcal{H}) \right) + \mathcal{B}, & \text{in }]x_1, x_2[& \times [t_i, t_f] \\ \mathcal{H} \geq 0, & \text{in }]x_1, x_2[& \times [t_i, t_f] \\ \mathcal{H} = 0, & \text{on } x = x_1, x_2 & \times [t_i, t_f] \\ \mathcal{H} = s_i - b, & \text{in }]x_1, x_2[& \times \{t_i\}, \end{array} \right. \quad (2.32)$$

where the diffusivities (2.21) specialize to

$$\mathcal{D}_f = \Gamma \mathcal{H}^{n+2} \left| \frac{\partial s}{\partial x} \right|^{n-1} \quad \text{and} \quad \mathcal{D}_s = \Gamma_s \mathcal{H}^{n+1} \left| \frac{\partial s}{\partial x} \right|^{n-1}. \quad (2.33)$$

2.4 Surface mass-balance

There exists many ways to model the surface mass-balance \mathcal{B} [61, 34, 62]. The following altitude-dependent, four-parameter expression is considered throughout this thesis:

$$\mathcal{B}(x, y, t) = \frac{m_1 + m_2}{2} \left(s(x, y, t) - z_{\text{ELA}} \right) - \sqrt{\left(\frac{m_1 - m_2}{2} \left(s(x, y, t) - z_{\text{ELA}} \right) \right)^2 + \varepsilon}, \quad (2.34)$$

where z_{ELA} is the equilibrium line altitude (ELA), m_1 and m_2 , $m_1 \geq m_2$, are the melting rates below and above the ELA respectively, and $\varepsilon \geq 0$ is a regularization parameter introduced to make it continuously differentiable. Such a mass balance parameterization describes ice accumulation above and ice ablation below the ELA, with rates m_2 and m_1 respectively (see Figures 2.2 and 2.7) and corresponds to an annual average of the precipitations occurring on the glacier [92]. If $m_1 \leq m_2$, then the roles of the rates are swapped. Surface mass-balance (2.34) is widely used among the glaciologists' community with $\varepsilon = 0$. The regularization parameter is here introduced in order to make the surface mass-balance differentiable with respect to all its parameters.

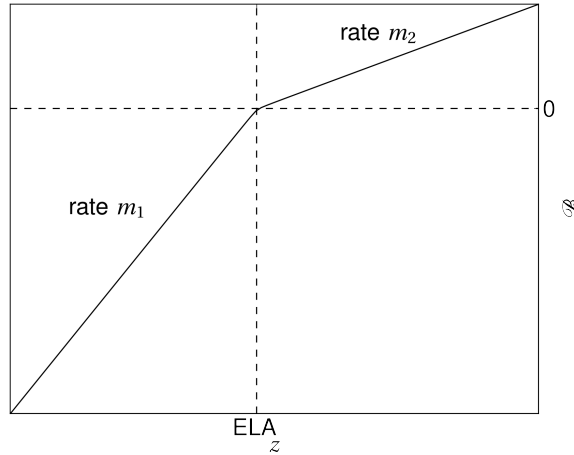


Figure 2.7 – Surface mass-balance function \mathcal{B} defined by (2.34) with $\varepsilon = 0.1$. The ELA corresponds to the surface altitude where $\mathcal{B} = 0$. Above this altitude, ice accumulates with a rate m_2 . Below it, ice melts with rate m_1 (see also Figure 2.2).

2.5 Conclusion

This chapter describes the two processes arising in ice flow dynamics, namely the internal ice mass and momentum conservation, governed by the Stokes equations, and the free surface evolution, ruled by the ice thickness equation. Their approximation for shallow glaciers is formulated where the ice velocity can be expressed analytically. In this thesis, either the coupling of the Stokes equations (2.8) with the VOF equation (2.27), or the shallow ice equations (2.32) and (2.31) in two and three space dimensions are considered. The next chapter deals with their numerical solving.

3 Forward numerical models

This chapter is devoted to the numerical methods aiming at solving the Stokes equations (2.8), the volume of fluid equation (2.27), and the shallow ice equations (2.31) and (2.32). First, the finite-element discretization of (2.8) is presented. Next, the time-splitting scheme used to solve the volume of fluid equation (2.27) is introduced. Then, the finite-difference schemes that discretize the shallow ice equations (2.31)-(2.32) are exposed. Finally, all these numerical methods are validated in the following way: First, the shallow ice schemes are applied to the Halfar glacier, whose evolution is known exactly [55, 56]. Then, a comparison of the Stokes to the shallow ice solutions is performed on sliding ice caps, before the confrontation of both solutions on four mountain glaciers in the Swiss Alps.

3.1 Methodology

Initially, the glacier's geometry is delimited by the bedrock and surface topographies b and s_i respectively. The objective is to compute the evolution of the glacier from time t_i to time t_f . In the forward problem solved numerically in this chapter, the bedrock topography is known while the ice thickness, therefore the surface topography, is unknown, except at initial time. The next chapters deal with the inverse problem where the surface topography is known while the ice thickness, therefore the bedrock topography in this case, is unknown.

In order to solve the forward problem, time is discretized. Let $\Delta t > 0$ be some time step and $t^\ell = t_i + \ell \Delta t$, $0 \leq \ell \leq M + 1$, such that $t^0 = t_i$ and $t^{M+1} = t_f$. Then, depending on the ice flow approximation, the solving process is different.

3.1.1 Stokes ice flow

At time iteration ℓ , three unknowns are sought, namely the approximations p^ℓ , \mathbf{u}^ℓ , and $\mathcal{H}^{\ell+1}$ of the ice pressure, velocity, and thickness respectively. The ice volume Ω^ℓ is first fixed and an approximation \mathbf{u}^ℓ of the ice velocity inside it is computed thanks to the Stokes equations (2.8) or its shallow ice approximations (2.31) and (2.32). In the case of a Stokes ice flow, the system of equations must be solved for the ice velocity and pressure. A finite-element method is advocated for this purpose with a mesh that covers the entirety of cavity Λ to take into account the free surface character of the problem in a simple way [79, 66].

Once the Stokes ice velocity is known, ice is transported according to the volume of fluid equation (2.27). Let φ^ℓ be the approximation of the VOF function φ at iteration ℓ . The VOF equation is solved on a mesh of cubic cells, with a forward characteristic method, whose numerical diffusion and compression are reduced by means of specific algorithms. The VOF equation calculates $\varphi^{\ell+1}$ from φ^ℓ and \mathbf{u}^ℓ . From $\varphi^{\ell+1}$, the new ice volume $\Omega^{\ell+1}$ is deduced, inside which the ice velocity $\mathbf{u}^{\ell+1}$ is to be computed, and so on [79, 66].

3.1.2 Shallow ice flow

At time iteration ℓ , the only unknown is the ice thickness $\mathcal{H}^{\ell+1}$. The analytical expressions (2.13)–(2.15) of the ice velocity are discretized on a finite-difference grid of the glacier map domain Ω_\perp [2]. Then, the diffusion equation (2.31) is solved with a finite-difference scheme in order to obtain the glacier's thickness \mathcal{H} .

The Stokes ice flow is described in the next section.

3.2 Stokes ice flow

3.2.1 Ice velocity computation

Let $\Omega^\ell \subset \Lambda$ be the current ice domain inside which a velocity has to be determined, *i.e.* the set of points in Λ such that the VOF function $\varphi^\ell \geq \frac{1}{2}$. The following Banach spaces are involved in the weak formulation of (2.8) [95, 97, 66]:

$$V^\ell = \left\{ \mathbf{v} \in \left[W^{1, \frac{n+1}{n}}(\Omega^\ell) \right]^3 \mid \mathbf{v} = 0 \text{ on } \Gamma_b^{\text{NS}} \right\} \quad (3.1)$$

$$Q^\ell = L^{n+1}(\Omega^\ell), \quad (3.2)$$

where n is Glen's flow law exponent that defines the behavior of the ice viscosity (see equation (2.4)). Usually, the condition $\mathbf{v} \cdot \mathbf{n} = 0$ on boundary Γ_b^S is part of space V^ℓ . For practical reasons, the velocity's normal component is penalized according to

$$\mathbf{u}^\ell \cdot \mathbf{n} = \varepsilon_{\text{FEM}} \left((\sigma^\ell \cdot \mathbf{n}) \cdot \mathbf{n} \right), \quad (3.3)$$

where $\varepsilon_{\text{FEM}} > 0$ is a regularization constant that is chosen as small as possible [87]. The weak formulation of (2.8) is [97, 66]: *find* $(\mathbf{u}^\ell, p^\ell) \in V^\ell \times Q^\ell$ *such that*:

$$\begin{cases} \int_{\Omega^\ell} \eta(\mathbf{u}^\ell) D(\mathbf{u}^\ell) : D(\mathbf{v}) + \sum_{l=1,2} \int_{\Gamma_b^S} \alpha(\mathbf{u}^\ell) (\mathbf{u}^\ell \cdot \mathbf{t}_l) (\mathbf{v} \cdot \mathbf{t}_l) - \frac{1}{\varepsilon_{\text{FEM}}} \int_{\Gamma_b^S} (\mathbf{u}^\ell \cdot \mathbf{n}) (\mathbf{v} \cdot \mathbf{n}) - \int_{\Omega^\ell} p^\ell \nabla \cdot \mathbf{v} = \rho \int_{\Omega^\ell} \mathbf{g} \cdot \mathbf{v}, & \forall \mathbf{v} \in V^\ell \\ \int_{\Omega^\ell} q \nabla \cdot \mathbf{u}^\ell = 0, & \forall q \in Q^\ell, \end{cases} \quad (3.4)$$

where \mathbf{t}_l , $l = 1, 2$, are the tangent vectors to Γ_b^S (see equation (2.8)₅), η is the viscosity defined by (2.4), α the sliding coefficient defined by (2.9), and

$$D(\mathbf{u}) : D(\mathbf{v}) = \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j}. \quad (3.5)$$

Let \mathcal{T}_H be a conforming triangulation of the cavity Λ that contains the ice domain Ω^ℓ . Figure 3.1 shows an example triangulation of Silvretta glacier with tetraedral, finite-elements. The cavity is chosen in such a way that it contains Ω^ℓ at all time iterations ℓ , so that the mesh remains unchanged during the whole glacier evolution. Hence, Λ consists in all the finite-elements depicted by Figure 3.1, of any color. The Stokes equations are solved on the

white elements only (see Figure 3.1), which are defined by the VOF equation. From one iteration to the other, these elements are therefore not the same due to the surface topography evolution. Basically, the VOF equation defines a new domain after each time iteration. The current ice domain Ω^ℓ is defined by all the finite-elements that are filled with ice. A finite-element said to be filled with ice and therefore called “active” if at least one of its nodes is filled with ice, *i.e.* its volume of fluid function is such that $\varphi^\ell \geq \frac{1}{2}$. All the algorithmic details behind this step are exposed rigorously in [79, 66]. In this thesis, weak formulation (3.4) is discretized with $\mathbb{P}_1 - \mathbb{P}_1$, tetraedral, finite-elements of size H and a Galerkin least squares (GLS) stabilization [38].

Consider the finite-dimensional spaces

$$V_H^\ell = V^\ell \cap \left\{ \mathbf{v}_H \in [\mathcal{C}^0(\overline{\Omega^\ell})]^3 : \mathbf{v}_H|_K \in [\mathbb{P}_1]^3, \forall K \in \mathcal{T}_H \right\} \quad (3.6)$$

$$Q_H^\ell = Q^\ell \cap \left\{ q_H \in \mathcal{C}^0(\overline{\Omega^\ell}) : q_H|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_H \right\} \quad (3.7)$$

of piecewise \mathbb{P}_1 polynomials on \mathcal{T}_H . In these spaces, the ice velocity and pressure are approximated by the linear combinations

$$\mathbf{u}_H^\ell(x, y, z) = \sum_{i=1}^{N^\ell} \mathbf{u}_i^\ell \Phi_i(x, y, z) \quad \text{and} \quad p_H^\ell(x, y, z) = \sum_{j=1}^{N^\ell} p_j^\ell \Phi_j(x, y, z), \quad (3.8)$$

where the functions Φ_i , $i = 1, 2, \dots, N^\ell$, are the \mathbb{P}_1 -basis functions of the finite-element space and the summations run over the N^ℓ nodes constituting all the active finite-elements.

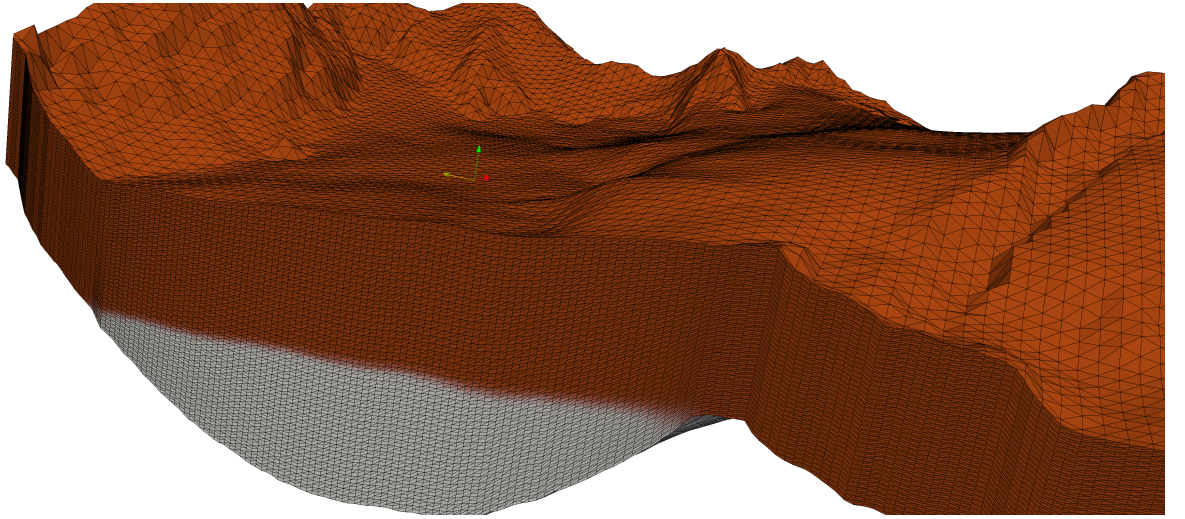


Figure 3.1 – Cut of the mesh used to solve the non-linear Stokes problem with finite-elements on Silvretta glacier. The cavity Λ consists in the volume spanned by all the brown and white tetraedra. Volume Ω^ℓ is represented by the white tetraedra. At this iteration, the Stokes problem is solved on the white elements only. The brown elements are there to take into account the future evolution of the ice surface, so as to keep a fixed mesh in the cavity during the whole glacier’s evolution.

Then, the discretized weak formulation reads

$$\left\{ \begin{array}{l} \int_{\Omega^\ell} \eta(\mathbf{u}_H^\ell) D(\mathbf{u}_H^\ell) : D(\mathbf{v}_H) + \sum_{l=1,2} \int_{\Gamma_b^s} \alpha(\mathbf{u}_H^\ell) (\mathbf{u}_H^\ell \cdot \mathbf{t}_l) (\mathbf{v}_H \cdot \mathbf{t}_l) - \frac{1}{\varepsilon_{\text{FEM}}} \int_{\Gamma_b^s} (\mathbf{u}_H^\ell \cdot \mathbf{n}) (\mathbf{v}_H \cdot \mathbf{n}) \\ - \int_{\Omega^\ell} p_H^\ell \nabla \cdot \mathbf{v}_H + \int_{\Omega^\ell} \tau_2 \nabla \cdot \mathbf{u}_H^\ell \nabla \cdot \mathbf{v}_H^\ell = \rho \int_{\Omega^\ell} \mathbf{g} \cdot \mathbf{v}_H, \quad \forall \mathbf{v}_H \in V_H, \\ \int_{\Omega^\ell} q_H \nabla \cdot \mathbf{u}_H^\ell + \int_{\Omega^\ell} \tau_1 \nabla p_H^\ell \cdot \nabla q_H = \rho \int_{\Omega^\ell} \tau_1 \mathbf{g} \cdot \nabla q_H, \quad \forall q_H \in Q_H^\ell, \end{array} \right.$$

where τ_1 and τ_2 are stabilization parameters [79, Chapter 4] that are explicited in Appendix A.1. This non-linear problem in the velocity and pressure is solved iteratively with a fixed-point procedure [66] following Algorithm 1. The implementation details are given in Appendix A.1.

This is all summarized in Algorithm 1, where $\delta > 0$ is the fixed-point tolerance.

Algorithm 1 Computation of the Stokes ice velocity \mathbf{u}^ℓ

```

set geometry  $\Omega^\ell$  from the VOF function  $\varphi^\ell$  according to definition (2.26)
set  $\mathbf{X}^0 = \mathbf{u}^{\ell-1}$  and  $Y^0 = p^{\ell-1}$ 
while  $\|\mathbf{X}^{i+1} - \mathbf{X}^i\| > \delta$  and  $|Y^{i+1} - Y^i| > \delta$  do
    build the linear system corresponding to (3.9) with velocity  $\mathbf{X}^i$  and pressure  $Y^i$ 
    solve the linear system with a GMRES method and an ILU preconditioner for  $\mathbf{X}^{i+1}, Y^{i+1}$ 
end while
set  $\mathbf{u}^\ell$  and  $p^\ell$  with the converged values of  $\mathbf{X}^i$  and  $Y^i$ 
    
```

3.2.2 Ice transport

The free surface can be captured with the help of the volume of fluid (VOF) equation (2.27), which is coupled to the Stokes equations (2.8) [102, 11, 21, 79]. At this point, the approximation \mathbf{u}^ℓ of the ice velocity is known and the aim is to compute the surface obtained after ice transport according to (2.27). Cavity Λ , the reunion of all the white and brown finite-elements represented in Figure 3.1, is in its turn enclosed in a parallelepipedic box filled with small, isotropic cubic cells of size $h \approx \frac{H}{5}$. Each cell located in Ω^ℓ is filled with fluid (white region of Figure 3.1). The values φ^ℓ of the volume fraction of ice have already been computed during the previous time iteration. Now, the values $\varphi^{\ell+1}$ are computed. The finite-element velocity \mathbf{u}^ℓ is linearly interpolated onto the VOF cells and the VOF equation is solved on these cells with the following time-splitting scheme [45]: First, solve the transport equation

$$\frac{\varphi^{\ell+\frac{1}{2}} - \varphi^\ell}{\Delta t} + \mathbf{u}^\ell \cdot \nabla \varphi^\ell = 0 \quad (3.10)$$

with a forward characteristic method [113, 80, 79] that updates the VOF function according to

$$\varphi^{\ell+\frac{1}{2}}(\mathbf{x}) = \varphi^\ell(\mathbf{x} + \Delta t \mathbf{u}^\ell(\mathbf{x})), \quad (3.11)$$

where \mathbf{x} is the position of each cell. This basically corresponds to move the cells following the ice velocity and then projecting the fluid contained in the cell onto the neighboring cells located near the target position. A simple linear interface calculation (SLIC) algorithm [24, 89] is then applied in order to attenuate numerical diffusion effects, while cell compression effects are countered by a decompression algorithm [69]. All the relevant implementation details are given in [41, 42, 69]. Then, solve the explicit-in-time surface mass-balance equation, following [69, 66]:

$$\frac{\varphi^{\ell+1} - \varphi^{\ell+\frac{1}{2}}}{\Delta t} = \mathcal{B}^{\ell+\frac{1}{2}} \delta_{\Gamma_s^{\ell+\frac{1}{2}}}. \quad (3.12)$$

Finally, since the values $\mathcal{B}^{\ell+\frac{1}{2}}$ can be negative, the constraint that

$$\varphi^{\ell+1} \geq 0 \quad (3.13)$$

is enforced for each cell. The VOF values on the cells are then extrapolated onto the finite-element mesh, which defines a new ice volume $\Omega^{\ell+1}$ (*i.e.* an updated white region on Figure 3.1) inside which the ice velocity must be calculated again through the solving of (2.8) with the finite-element method of previous section, and so on. From $\varphi^{\ell+1}$, the updated surface topography is deduced from (2.29). The method has order one in time and space. Algorithm 2 summarizes the coupled process of computing the ice velocity and then updating the ice volume.

Algorithm 2 Glacier evolution in the case of a Stokes ice flow.

```

set glacier geometry with bedrock topography  $b$  and initial surface  $s_i$ 
set  $\Omega^0$  and  $\varphi^0$  with (2.26)
for  $\ell = 0, 1, 2, \dots, M$  do
    compute the finite-element ice velocity  $\mathbf{u}^\ell$  inside  $\Omega^\ell$  with Algorithm 1
    project the finite-element ice velocity  $\mathbf{u}^\ell$  onto the mesh of VOF cells [79, equation (4.24)]
    solve the VOF transport equation (3.10) with a forward characteristic method [79, Chapter 4, Section 2.1]
    apply the SLIC algorithm [79, Chapter 4, Section 2.2]
    apply the decompression algorithm [69]
    solve the surface mass-balance equation (3.12) with the constraint (3.13) [69, 66]
    project the VOF fluid onto the finite-element mesh to define the new ice volume  $\Omega^{\ell+1}$  [79, equation (4.10)]
end for
    
```

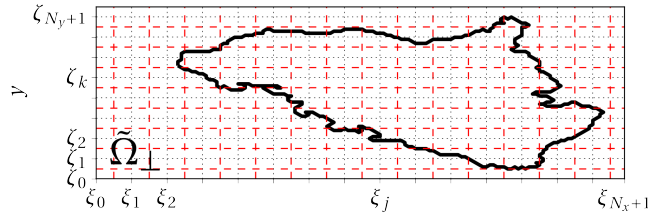


Figure 3.2 – Direct grid of the glacier’s map domain Ω_\perp (black dotted lines). The middle of each edge is marked with the red, dashed lines, which show the points considered for the centered evaluation of gradients.

3.3 Shallow ice flow

Centered-in-space, finite-difference schemes are advocated to solve (2.31). Let $\Delta x > 0$ be some space step and $(\xi_j, \zeta_k) = (j, k)\Delta x$, $0 \leq j \leq N_x + 1$, $0 \leq k \leq N_y + 1$, vertices of the so-called “direct grid” $\tilde{\Omega}_\perp$ in Ω_\perp (black dotted lines on Figure 3.2). In particular, $L_x = (N_x + 1)\Delta x$ and $L_y = (N_y + 1)\Delta x$.

Let us denote the approximation of the bedrock topography $b(\xi_j, \zeta_k)$, ice thickness $\mathcal{H}(\xi_j, \zeta_k, t^\ell)$, surface topography $s(\xi_j, \zeta_k, t^\ell)$, surface mass-balance $\mathcal{B}(\xi_j, \zeta_k, t^\ell)$, and ice velocity $\mathcal{U}(\xi_j, \zeta_k, t^\ell)$ at location (ξ_j, ζ_k) and time t^ℓ with b_{jk} , \mathcal{H}_{jk}^ℓ , $s_{jk}^\ell = b_{jk} + \mathcal{H}_{jk}^\ell$, \mathcal{B}_{jk}^ℓ , and \mathcal{U}_{jk}^ℓ . Denote by \mathbf{b} , \mathcal{H}^ℓ , \mathbf{s}^ℓ , \mathcal{B}^ℓ , and \mathcal{U}^ℓ the corresponding vectors. Moreover, let us write \mathbf{s}_i and \mathbf{s}_f the vectors of components $s_i(\xi_j, \zeta_k)$ and $(s_f)_{jk} = b_{jk} + \mathcal{H}_{jk}^{M+1}$, that are the glacier’s surface at initial and final times respectively. Recall that the initial surface topography s_i is known while s_f is computed with

the numerical scheme and that \mathcal{W}_{jk}^ℓ refers to the horizontal surface velocity defined by (2.22).

The solving process is again based on the equation's time splitting. First, the advection-diffusion equation is considered without source term. Then, an equation for the source term is solved. The finite-difference scheme, centered in space, discretizing (2.31), consists in initializing the ice thickness with

$$\mathcal{H}_{jk}^0 = s_i(\xi_j, \zeta_k) - b_{jk} \quad (3.14)$$

for $1 \leq j \leq N_x$, $1 \leq k \leq N_y$; then, for $0 \leq \ell \leq M$, first solve the advection-diffusion scheme [78, 17, 2, 18, 103]

$$\frac{\overline{\mathcal{H}}_{jk}^\ell - \mathcal{H}_{jk}^\ell}{\Delta t} = \mathcal{T}_{jk}^\ell, \quad (3.15)$$

where \mathcal{T}_{jk}^ℓ is the discretization of $\nabla \cdot (\mathcal{D} \nabla s)$; second, solve the mass-balance equation

$$\frac{\overline{\overline{\mathcal{H}}}_{jk}^\ell - \overline{\mathcal{H}}_{jk}^\ell}{\Delta t} = \vartheta(\overline{\mathcal{H}}_{jk}) \overline{\overline{\mathcal{B}}}_{jk}^\ell, \quad (3.16)$$

where ϑ is the Heaviside function

$$\vartheta(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.17)$$

and $\overline{\overline{\mathcal{B}}}_{jk}^\ell$ is the surface mass-balance (2.34) evaluated with $\overline{s}_{jk}^\ell = b_{jk} + \overline{\overline{\mathcal{H}}}_{jk}^\ell$; finally, project the thickness such that it is positive:

$$\mathcal{H}_{jk}^{\ell+1} = \left[\overline{\overline{\mathcal{H}}}_{jk}^\ell \right]^+, \quad (3.18)$$

where $[\cdot]^+$ is defined by (2.11). In the semi-implicit discretization, the right-hand side of (3.15) writes

$$\mathcal{T}_{jk}^\ell = \frac{1}{\Delta x} \left[\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{\overline{s}_{j+1,k}^\ell - \overline{s}_{jk}^\ell}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{\overline{s}_{jk}^\ell - \overline{s}_{j-1,k}^\ell}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{\overline{s}_{j,k+1}^\ell - \overline{s}_{jk}^\ell}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{\overline{s}_{jk}^\ell - \overline{s}_{j,k-1}^\ell}{\Delta x} \right], \quad (3.19)$$

where

$$\mathcal{D}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \left(\Gamma \mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + (\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} \right) \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n+1,n-1}(\ell) \quad (3.20)$$

$$\mathcal{D}_{j+\frac{1}{2},k}^\ell = \frac{1}{2} \left(\mathcal{D}_{j+\frac{1}{2},k-\frac{1}{2}}^\ell + \mathcal{D}_{j+\frac{1}{2},k+\frac{1}{2}}^\ell \right) \quad (3.21)$$

$$\mathcal{D}_{j,k+\frac{1}{2}}^\ell = \frac{1}{2} \left(\mathcal{D}_{j-\frac{1}{2},k+\frac{1}{2}}^\ell + \mathcal{D}_{j+\frac{1}{2},k+\frac{1}{2}}^\ell \right) \quad (3.22)$$

discretize the diffusivity (2.20)-(2.21) and

$$\mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{p,q}(\ell) = \left(\mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell \right)^p \left(\alpha_{j-\frac{1}{2},k-\frac{1}{2}}^\ell \right)^q, \quad (3.23)$$

$$\mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \frac{\mathcal{H}_{j-1,k-1}^\ell + \mathcal{H}_{j,k-1}^\ell + \mathcal{H}_{j-1,k}^\ell + \mathcal{H}_{j,k}^\ell}{4}, \quad (3.24)$$

$$(\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} = (\mathcal{C} \rho g)^n \frac{[z_{sl} - b]_{j-1,k-1}^+ + [z_{sl} - b]_{j,k-1}^+ + [z_{sl} - b]_{j-1,k}^+ + [z_{sl} - b]_{j,k}^+}{4}, \quad (3.25)$$

$$\alpha_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \left[\left(\frac{s_{jk}^\ell - s_{j-1,k}^\ell + s_{j,k-1}^\ell - s_{j-1,k-1}^\ell}{2\Delta x} \right)^2 + \left(\frac{s_{jk}^\ell - s_{j,k-1}^\ell + s_{j-1,k}^\ell - s_{j-1,k-1}^\ell}{2\Delta x} \right)^2 \right]^{\frac{1}{2}}. \quad (3.26)$$

Note that $\alpha_{j-\frac{1}{2},k-\frac{1}{2}}^\ell$ is not the sliding coefficient defined by (2.9) but the discretization of $\|\nabla s\|$ at position $(j - \frac{1}{2}, k - \frac{1}{2})\Delta x$ and time t^ℓ instead. The half-integer indices refer to the mid-points $(j + \frac{1}{2}, k + \frac{1}{2})\Delta x$ that are artificially introduced for the centered evaluation of the involved gradients (see dashed, red lines on Figure 3.2).

In turn, the finite-difference scheme, centered in space, implicit in time, discretizing (2.31), consists in applying scheme (3.14) – (3.18) with

$$\begin{aligned} \mathcal{T}_{jk}^\ell = \frac{1}{2\Delta x} & \left[\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{s_{j+1,k}^\ell - s_{jk}^\ell}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{s_{jk}^\ell - s_{j-1,k}^\ell}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{s_{j,k+1}^\ell - s_{jk}^\ell}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{s_{jk}^\ell - s_{j,k-1}^\ell}{\Delta x} \right] \\ & + \frac{1}{2\Delta x} \left[\overline{\mathcal{D}}_{j+\frac{1}{2},k}^\ell \frac{\bar{s}_{j+1,k}^\ell - \bar{s}_{jk}^\ell}{\Delta x} - \overline{\mathcal{D}}_{j-\frac{1}{2},k}^\ell \frac{\bar{s}_{jk}^\ell - \bar{s}_{j-1,k}^\ell}{\Delta x} + \overline{\mathcal{D}}_{j,k+\frac{1}{2}}^\ell \frac{\bar{s}_{j,k+1}^\ell - \bar{s}_{jk}^\ell}{\Delta x} - \overline{\mathcal{D}}_{j,k-\frac{1}{2}}^\ell \frac{\bar{s}_{jk}^\ell - \bar{s}_{j,k-1}^\ell}{\Delta x} \right], \end{aligned} \quad (3.27)$$

where $\overline{\mathcal{D}}$ is the diffusivity \mathcal{D} evaluated with $\overline{\mathcal{H}}$. Theoretically, this implicit scheme is more likely to suffer from convergence issues, as it is solved with a Newton method at each time iteration and hence a good initial guess must be provided each time. However, in all the computations performed throughout this thesis, no such problems arose.

For the sake of simplicity, especially in the coming up stationary and quasi-stationary approximations (Chapter 5), equations (3.15) and (3.16) are merged into the following

$$\begin{aligned} \frac{\mathcal{H}_{jk}^{\ell+1} - \mathcal{H}_{jk}^\ell}{\Delta t} = \frac{1}{\Delta x} & \left[\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{s_{j+1,k}^{\ell+1} - s_{jk}^{\ell+1}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{s_{jk}^{\ell+1} - s_{j-1,k}^{\ell+1}}{\Delta x} \right. \\ & \left. + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{s_{j,k+1}^{\ell+1} - s_{jk}^{\ell+1}}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{s_{jk}^{\ell+1} - s_{j,k-1}^{\ell+1}}{\Delta x} \right] + \mathcal{B}_{jk}^\ell \end{aligned} \quad (3.28)$$

in the semi-implicit time discretization and

$$\begin{aligned} \frac{\mathcal{H}_{jk}^{\ell+1} - \mathcal{H}_{jk}^\ell}{\Delta t} = \frac{1}{2\Delta x} & \left[\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{s_{j+1,k}^\ell - s_{jk}^\ell}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{s_{jk}^\ell - s_{j-1,k}^\ell}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{s_{j,k+1}^\ell - s_{jk}^\ell}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{s_{jk}^\ell - s_{j,k-1}^\ell}{\Delta x} \right] \\ & + \frac{1}{2\Delta x} \left[\mathcal{D}_{j+\frac{1}{2},k}^{\ell+1} \frac{s_{j+1,k}^{\ell+1} - s_{jk}^{\ell+1}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^{\ell+1} \frac{s_{jk}^{\ell+1} - s_{j-1,k}^{\ell+1}}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^{\ell+1} \frac{s_{j,k+1}^{\ell+1} - s_{jk}^{\ell+1}}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^{\ell+1} \frac{s_{jk}^{\ell+1} - s_{j,k-1}^{\ell+1}}{\Delta x} \right] + \mathcal{B}_{jk}^\ell \end{aligned} \quad (3.29)$$

in the implicit time discretization. The condition $\mathcal{H}_{jk}^\ell \geq 0$ is taken into account during the solving of (3.28) or (3.29) by forcing the thickness to be positive after each iteration, in both time discretizations.

From the computed thickness, the horizontal surface velocity can be expressed on the staggered grid with

$$u_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \left(\frac{n+2}{n+1} \Gamma \mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + (\Gamma s)_{j-\frac{1}{2},k-\frac{1}{2}} \right) \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell) \quad (3.30)$$

and (2.22) is approached with

$$\mathcal{U}_{jk}^\ell = \frac{1}{4} \left(u_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + u_{j+\frac{1}{2},k-\frac{1}{2}}^\ell + u_{j-\frac{1}{2},k+\frac{1}{2}}^\ell + u_{j+\frac{1}{2},k+\frac{1}{2}}^\ell \right). \quad (3.31)$$

Finally, in all the validations and algorithms throughout this thesis, the numerical schemes applied to solve the two-dimensional shallow ice equation (2.32) are the two-dimensional specializations of (3.28) in the semi-implicit

and (3.29) in the implicit time discretizations. In these validations, no sliding is taken into account, so as to keep the equations as simple as possible. This amounts to

$$\begin{cases} \frac{\mathcal{H}_j^{\ell+1} - \mathcal{H}_j^\ell}{\Delta t} = \mathcal{T}_j^\ell + \mathcal{B}_j^\ell, & 1 \leq j \leq N_x, \quad \ell > 0, \\ \mathcal{H}_j^0 & = s_i(\xi_j) - b_j, \quad 1 \leq j \leq N_x \\ \mathcal{H}_j^\ell & \geq 0, \quad 1 \leq j \leq N_x, \quad 0 < \ell \leq M \\ \mathcal{H}_0^\ell = \mathcal{H}_{N_x+1}^\ell & = 0, \quad \ell \geq 0, \end{cases} \quad (3.32)$$

where

$$\mathcal{T}_j^\ell = \frac{1}{\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \right] \quad (3.33)$$

in the semi-implicit time discretization, where

$$(\mathcal{D}_f)_{j-\frac{1}{2}}^\ell = \Gamma \left(\frac{\mathcal{H}_{j-1}^\ell + \mathcal{H}_j^\ell}{2} \right)^{n+2} \left| \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} \right|^{n-1} \quad (3.34)$$

is the two-dimensional specialization of (3.20). In the implicit time discretization, the right-hand side of (3.32) is

$$\mathcal{T}_j^\ell = \frac{1}{2\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} \right] + \frac{1}{2\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^{\ell+1} \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^{\ell+1} \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \right]. \quad (3.35)$$

The implementation details of all the schemes presented above are specified in Appendix A.2.

3.4 Validation

The Stokes solver was validated in [79, 66, 69, 67]. Here, only the shallow ice solver shall be validated. First, let us consider the SIA fundamental solution of problem (2.31) for $n = 3$, $\mathcal{B} = 0$, no sliding, and a flat bed, known as the Halfar thickness [55, 56]:

$$\mathcal{H}_{\text{Halfar}}(r, t) = \mathcal{H}_0 \left(\frac{t_0}{t} \right)^{\frac{1}{9}} \left[1 - \left(\frac{r}{R_0} \left(\frac{t_0}{t} \right)^{\frac{1}{18}} \right)^{\frac{4}{3}} \right]^{\frac{3}{7}}, \quad (3.36)$$

where $r = \sqrt{x^2 + y^2}$, t_0 is the characteristic time scale

$$t_0 = \frac{1}{18\Gamma} \left(\frac{7}{4} \right)^3 \frac{R_0^4}{\mathcal{H}_0^7}, \quad (3.37)$$

and \mathcal{H}_0 , R_0 are the central thickness and ice cap radius at time $t = t_0$ (see Figure 3.3).

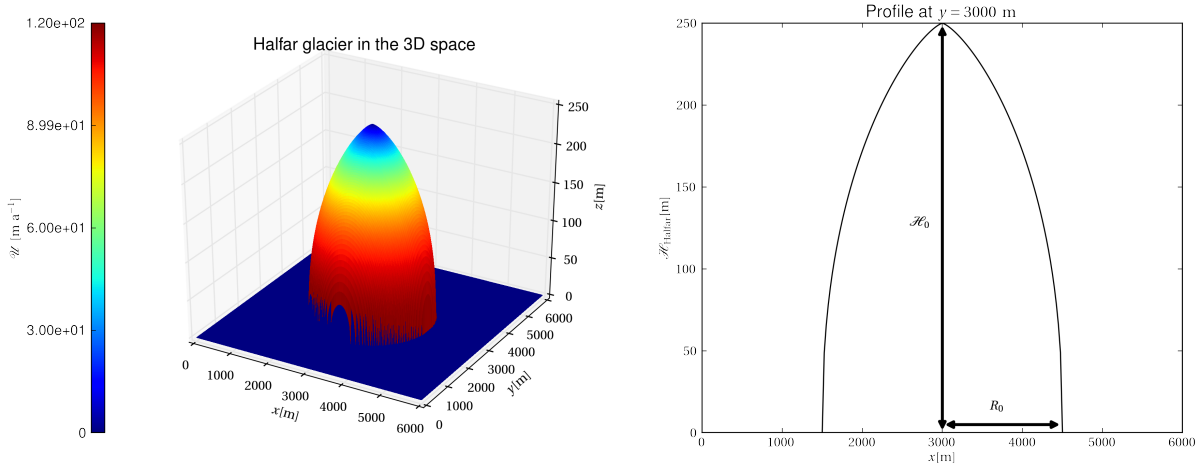


Figure 3.3 – Halfar fundamental solution of the shallow ice equation (2.31) at time $t = t_0$ when $n = 3$, $\mathcal{B} = 0$, the bedrock topography is flat, and no sliding occurs. It is an ice cap with maximum thickness \mathcal{H}_0 and radius R_0 . In the validation presented here, the parameters are set to $R_0 = 1500$ m, $\mathcal{H}_0 = 250$ m, and $A = 0.1 \text{ bar}^{-3} \cdot \text{a}^{-1}$. Left: a three-dimensional representation of the ice cap where the colors represent \mathcal{U} and the dome the glacier's initial surface. Right: Profile of the Halfar glacier along the curve $y = 3000$ m.

Both schemes have theoretical order two in space. However, the approximation of the glacier's boundaries has order one. Hence, the overall approximation has order one. This can be improved by using either finite-elements or sophisticated, finite-difference techniques that take the boundaries into account more appropriately. Here, the above algorithms are considered as they are for the sake of simplicity. Moreover, a large precision on the shallow ice approximation is not useful, as it is already a rough approximation of the Stokes ice flow. Since their application mainly aims at inverse approaches and most literature on the topic that makes use of the SIA applies them in this way, the same holds here. Only scheme (3.27) is new and solved in the same way as its semi-implicit-in-time counterpart. Its introduction is due to its better stability. Indeed, as far as our numerical experiments can tell, both schemes satisfy a stability condition of the form $\Delta t < c \Delta x^p$, where c and p are scheme and geometry-dependent and $p \leq 2$. The stability condition for (3.27) is less restrictive than that for (3.19). Numerical experiments on ice caps of various inclination angles give $p \approx 2$ in the semi-implicit scheme and $p \approx 1$ in the implicit case, but these are only indicative values that depend on the glacier's geometry. With decreasing space and time steps, the error layer around the boundaries becomes smaller and smaller with both schemes (see Figures 3.4 and 3.5). The error is largest near the boundaries, as expected.

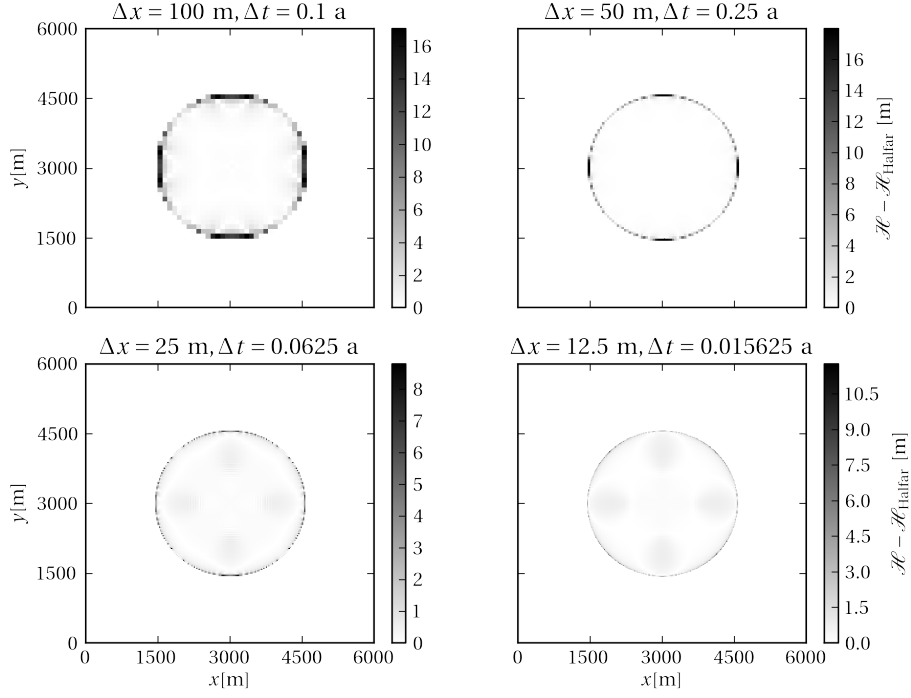


Figure 3.4 – Comparison between the Halfar fundamental solution (3.36) and the computed thickness with scheme (3.14)-(3.15)-(3.16)-(3.18) in the semi-implicit time discretization (3.19). The maximal errors are located near the glacier's boundaries. With the mesh size, the error layer around the boundaries becomes smaller.

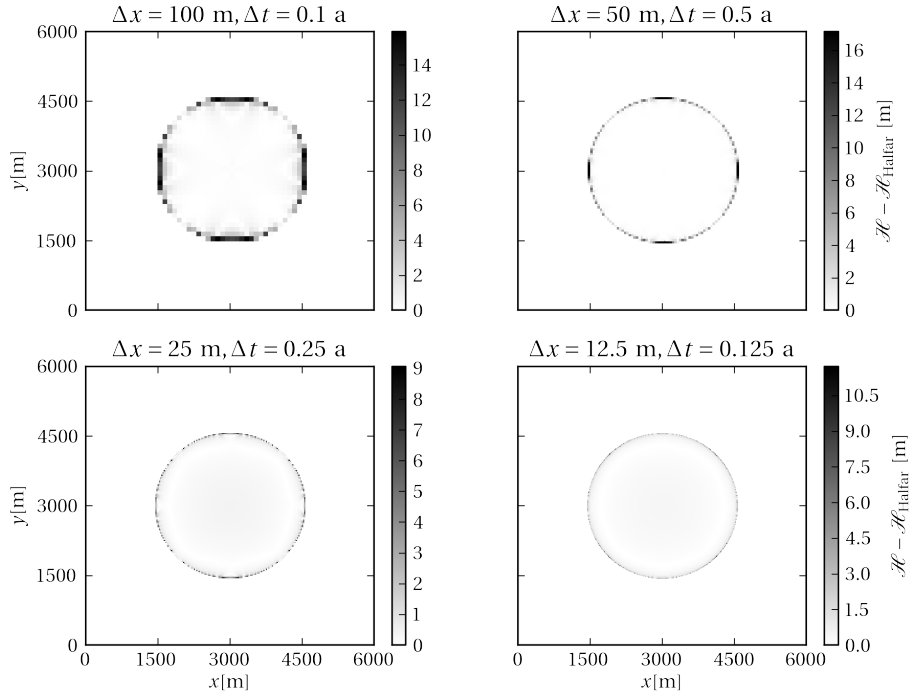


Figure 3.5 – Comparison between the Halfar fundamental solution (3.36) and the computed thickness with scheme (3.14)-(3.15)-(3.16)-(3.18) in the implicit time discretization (3.27). The results are analogous to those obtained in the semi-implicit time discretization, depicted by Figure 3.4.

Second, the sliding, inclined ice caps depicted by Figure 3.6 are considered. In all cases, the sliding is defined near the ice cap's tongue, except for zero inclination, where the sliding is defined uniformly on the boundaries. The sliding parameter is here set to the constant $C = 1 \text{ bar} \cdot \left(\frac{\text{a}}{\text{m}}\right)^{\frac{1}{3}}$. The Stokes approximation is solved on a structured, anisotropic, finite-element mesh of size $H = 25 \text{ m}$, while the VOF cells are transported on a structured mesh of cubic cells of size $h = 5 \text{ m}$ and time step $\Delta t = 0.5 \text{ a}$. The shallow ice approximation is solved with the implicit-in-time scheme (3.27) on a structured grid of size $\Delta x = 25 \text{ m}$ and with a time step $\Delta t = 0.01 \text{ a}$. The solution is far smoother in the shallow ice than in the Stokes approximation. A good agreement between the two solutions in every case can be seen, as far as the ice cap's thickness is thin enough, especially in the case $\alpha = 10^\circ$.

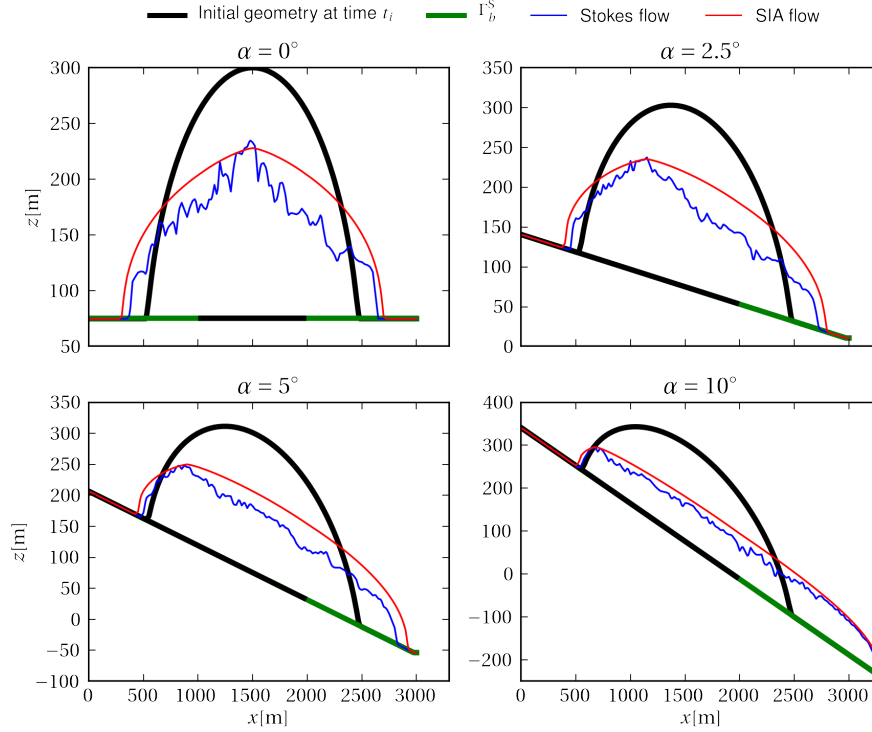


Figure 3.6 – Profile comparison between the three-dimensional Stokes and SIA solutions on sliding ice caps of inclination angles $\alpha = 0, 2.5, 5, 10^\circ$ and initial radius of extent 1000 m. In the SIA, scheme (3.14)-(3.15)-(3.16)-(3.18) in the implicit time discretization (3.27) is applied. The profile line shown here passes through the glacier's center. The solid, thick, black curves delineate the glaciers' initial geometries. The blue and red lines show the glaciers' surfaces after 10 a in the Stokes and shallow ice approximations respectively. The continuous, thick, green line shows the sliding domain Γ_b^S along the depicted profile lines. The thinner the glacier's final ice mass, the closer the shallow ice to the Stokes approximation.

Finally, four mountain glaciers located in the Swiss Alps are considered, whose measured ice thickness distributions are depicted by Figure 3.7 [34]. Figure 3.8 shows how different the shallow ice and Stokes approximations of ice flow are on these geometries. Basically, it can be seen that the estimations are usually significantly different. The glacier's surface is misestimated by an amount of about 30 m on the thin Gries, Silvretta, and Muragl glaciers. The shallow ice approximation has the tendency to overestimate the ice thickness a lot at the glaciers' tongues, especially on Muragl and Rhône glaciers. It is slightly underestimated up-glacier in all cases. The biggest misestimation is observed on Rhône glacier, that is the deepest one of the set, and amounts to 160 m. This can also be seen on the profiles of Figure 3.6, where the misfit at the tongues is clearly evinced.

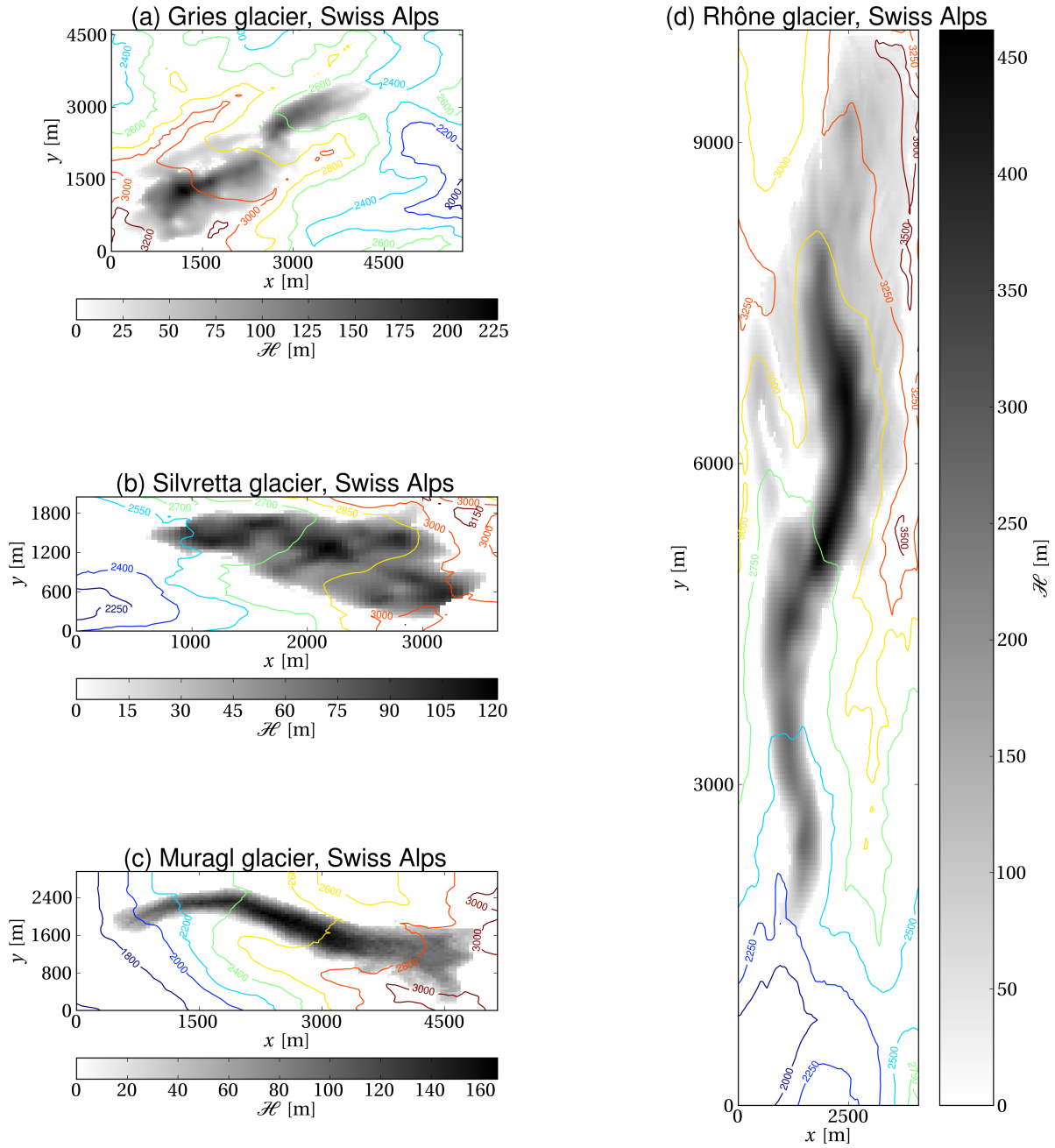


Figure 3.7 – Ice thickness distribution of Gries, Silvretta, Muragl, and Rhône glaciers in the Swiss Alps [34]. Gries, Silvretta and Muragl glaciers are much thinner than Rhône glacier, with 227, 120, 170, and 462 m maximal thickness respectively. The colored lines indicate the surface topography elevation and therefore the ice flow direction.

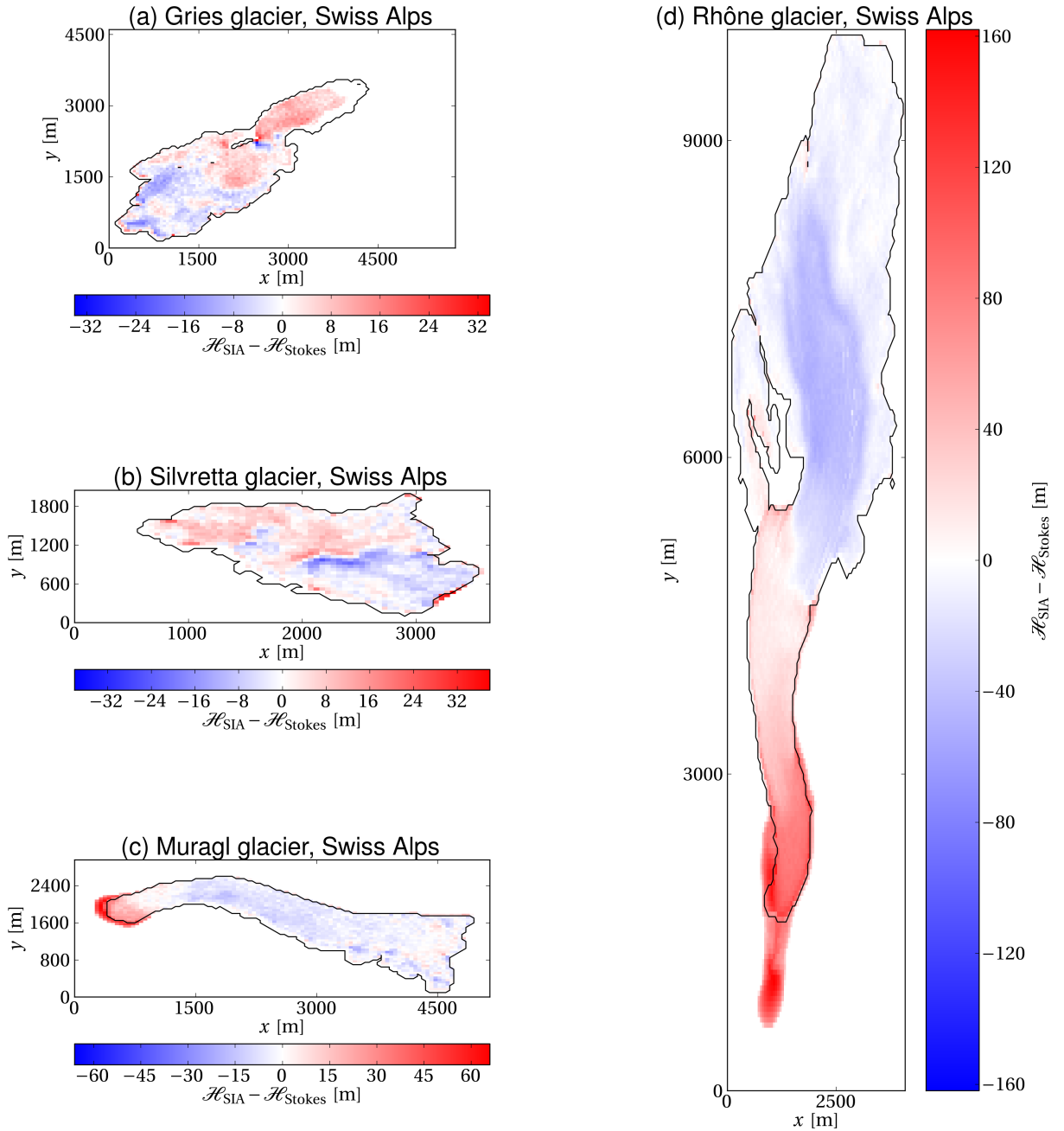


Figure 3.8 – Illustration of the difference $\mathcal{H}_{\text{SIA}} - \mathcal{H}_{\text{Stokes}}$ between the shallow and Stokes ice approximations on Gries, Silvretta, Muragl, and Rhône mountain glaciers, after an evolution of 10 a. Data for the numerical experiments come from [34]. The shallow ice evolution is performed with a finite-difference scheme, centered in space, implicit in time, with space and time steps $\Delta x = 25$ m and $\Delta t = 10^{-2}$ a. The Stokes evolution (2.8) is calculated with structured, tetraedral finite-elements of extent 25 m in the (x, y) -plane, 6.25 m along the z -axis, coupled with the VOF equation (2.27) solved on a structured grid of isotropic, cubic cells of size 5 m with a time step $\Delta t = 0.1$ a. The thick black curves delineate the contour of each glacier at initial time. In the red regions, the shallow ice overestimates the ice thickness, while it underestimates it in the blue regions. The SIA predicts too large thicknesses at the glaciers' tongues and underestimates thicknesses up-glacier compared to the Stokes approximation. The worst case is Rhône glacier where the maximal misfit is about 160 m.

3.5 Conclusion

A stabilized, $\mathbb{P}_1 - \mathbb{P}_1$, finite-element method is exposed that solves for the Stokes velocity and pressure according to equations (2.8) [79, 66, 80, 37]. Coupled with the latter method, a time-splitting scheme is presented that solves the volume of fluid equation (2.27) [45]. Through the transport of ice with the Stokes velocity, the VOF equation determines a new glacier surface topography. It is splitted into a transport and a surface mass-balance equation [69]. The transport equation is solved with a forward characteristic method [113, 80], complemented with a SLIC algorithm that reduces numerical diffusion [24, 89] and a decompression algorithm that reduces numerical compression of the VOF cells [79, 41, 42]. The surface mass-balance is solved by adding fluid to the top of each column of ice in the ice volume [69].

Because the computation of the full-Stokes solution is time and memory-consuming, especially in inverse approaches, where the forward model is run several times, the far simpler shallow ice approximation (2.31) is introduced. In this approximation, ice flow is essentially a simple, bed-parallel shear. Many finite-difference schemes are presented that are centered in space and either semi-implicit or implicit in time, in two and three dimensions of space. The two-dimensional formulations are very useful to perform fast numerical experiments and give insight into the more complicated, three-dimensional computations.

Finally, a validation of the presented numerical schemes is performed in three space dimensions. First, the finite-difference schemes aiming at the simulation of the shallow ice flows are confronted to the exact solution of the Halfar problem. The maximal errors are found on the glacier's boundaries, which is not surprising since their approximation has order one in space. Next, the Stokes and shallow ice approximations are compared on the time evolution of sliding ice caps of various inclinations. The more inclined the ice cap, the closer the shallow to the Stokes approximation, since the glacier is also shallower. Last, the differences in the evolution in both approximations of four mountain glaciers located in the Swiss Alps is examined. The two estimations are significantly different in all cases. Of course, they are closer in the shallowest glaciers of Silvretta and Muragl.

4 Sensitivity analysis

All the parameters the ice model depends on cumulate the drawbacks of both being laboriously measurable and describing in a possibly too simple way a glacier's dynamics. Addressed here is how critical it is to run a simulation with a wrong model parameter in forward approaches, when neither measurement nor model errors are involved in the process. Hence, in this chapter, the models are assumed perfect and therefore tested on synthetical geometries only. Various numerical experiments are presented that give a glimpse of the models' sensitivity to their parameters on synthetical data inspired from measurements on the four Swiss glaciers of Silvretta, Muragl, Gries, and Rhône [34].

Unless stated otherwise, the meshes used to solve the Stokes equations (2.8) consist of tetraedral, structured, anisotropic, finite-elements of size 25 m in the (x, y) -plane and 6.25 m along the z -axis. The VOF equation (2.27) in turn is solved on a structured mesh of isotropic, cubic cells of size 5 m and time step $\Delta t = 0.5$ a. The shallow ice equation is solved with the implicit scheme (3.27) on structured grids with space and time steps $\Delta x = 25$ m and $\Delta t = 0.01$ a respectively. The difference of magnitude in the time steps is due to the numerical schemes applied. Indeed, the stability condition of the shallow ice schemes are more restrictive than that of the VOF scheme. Moreover, for a given glacier, all the simulations performed here are compared to the simulation of a corresponding reference glacier that evolves from time $t_i = 0$ to time $t_f = 10$ a. Let us fix Glen's exponent to $n = 3$. As a reference, we choose the aforementioned glaciers with the bedrock topographies measured by [34] and the following model parameters:

$$\begin{aligned}
 A &= 0.1 & \text{bar}^{-3} \text{a}^{-1} \\
 m_1 = m_2 &= 0.01 & \text{a}^{-1} \\
 z_{\text{sl}} &= 2600 & \text{m} \\
 z_{\text{ELA}} &= 2800 & \text{m} \\
 \mathcal{C} &= 0.1 & \text{bar}^{-1} \text{a}^{-\frac{1}{3}} \\
 \mathcal{H}_{\text{max}} &= 120 & \text{m},
 \end{aligned} \tag{4.1}$$

where m_1 , m_2 , and z_{ELA} are the parameters of the surface mass-balance (2.34) and z_{sl} and \mathcal{C} are the parameters (2.10) of the sliding law. Although they are reasonable, these are not the values tuned to field data and should be considered as purely artificial, the purpose here being only to produce a reference time evolution for each glacier. Several experiments are performed here in both the Stokes and shallow ice approximations of ice flow. First, on the fixed geometry of Silvretta glacier, the influence of each of the parameters (4.1) is investigated, when all the

other ones are left unchanged. This is interesting to identify which ones of these parameters change the physics significantly. The simulation outputs are compared on different meshes and in both the Stokes and shallow ice approximations. Second, the influence of a bump added to the bedrock topographies of the aforementioned glaciers on various locations is studied. This is of prime importance for the algorithms presented in the next chapters based on the vertical bedrock-to-surface perturbation transfers.

4.1 Changing parameters on a fixed geometry

Figure 4.1 shows contour plots of Silvretta glacier. The coloured curves delineate some possible sliding and equilibrium line altitudes on the left and right subplots respectively. Originally, the glacier's maximal ice thickness is about 120 m. This first experiment consists in the simulation of the glacier's evolution over time frames of 1 and 10 a with different values of parameters (4.1). The resulting final surface topographies are then compared to the reference at the selected locations marked on Figure 4.1 with black crosses.

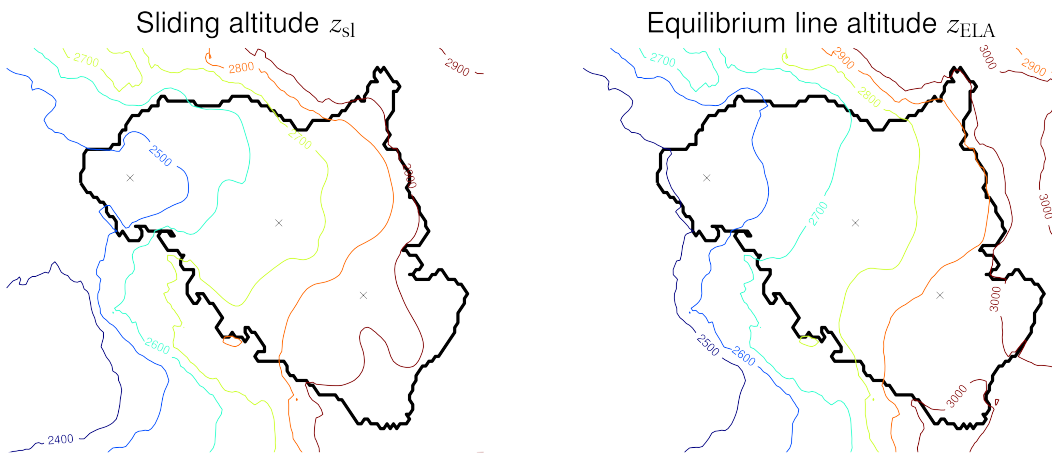


Figure 4.1 – Contour plots of Silvretta glacier, Swiss Alps, where the coloured lines represent the possible sliding altitude z_{sl} (left) and equilibrium line altitude z_{ELA} (right) arising in modelisations (2.10) and (2.34). The black crosses show the locations where comparisons are performed in the numerical experiments. The left-most cross is located down-glacier, the right-most one up-glacier, *i.e.* ice flows from the right to the left, as indicated by the contour lines. The contour lines for z_{sl} and z_{ELA} do not match because they depend on the bedrock and surface topographies respectively.

Figure 4.2 shows how the final surface is affected by the model parameters. The black dash-dotted lines mark the reference values (4.1), which correspond to zero surface discrepancy. Every time, the surface discrepancy is reported, on a different scale, because each of these parameters affects the simulation in a different manner. Hence, it is not physically relevant to choose the parameters' range in such a way that the resulting sensitivity in each case is comparable. In Figure 4.2, the parameters are chosen in a range that is reasonable for glaciers. The surface response changes with the location in the glacier domain. It is denoted with the labels “up-glacier”, “mid-glacier”, and “down-glacier” that correspond to the right-most, middle, and left-most crosses on Figure 4.1 respectively. Colored solid lines describe the sensitivities after a 1 a-evolution, while colored dotted lines those after a 10 a-evolution. Figure 4.2(b) depicts the results obtained when the glacier geometry is initialized with a thickness distribution that is solution of the Laplace equation

$$\Delta \mathcal{H} = \text{cst} \tag{4.2}$$

inside the glacier contour, where $\text{cst} \in \mathbb{R}$ is such a constant that the maximal glacier thickness \mathcal{H}_{\max} is some chosen

value, namely 100, 200, and 300 m. In the Figure, the case $\mathcal{H} = 120$ m corresponds to the reference geometry inspired from measurements.

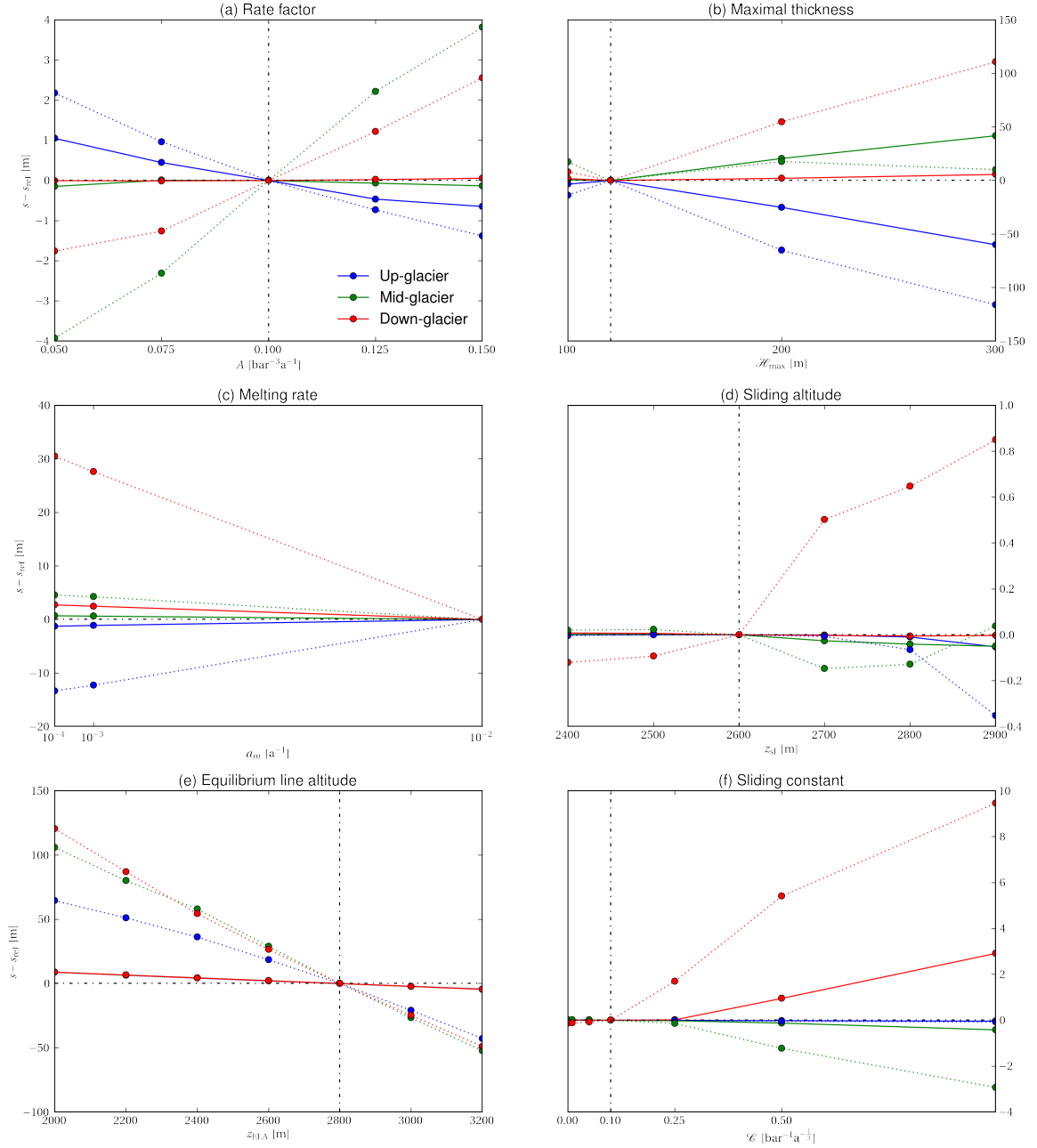


Figure 4.2 – Surface topography discrepancy obtained after simulation of the glacier’s Stokes evolution with different values of the model parameters. The black dash-dotted lines represent the reference values (4.1) in each case, while the solid coloured lines the discrepancies obtained over a time frame of 1 a and the dotted coloured lines those over 10 a. Blue, green, and red colors represent results obtained up, mid, and down-glacier, *i.e.* on the right-most, middle, and left-most crosses on Figure 4.1 respectively. The sensitivity is higher with the size of the time frame, the highest being attained with ice thickness (b) and mass-balance perturbations (e).

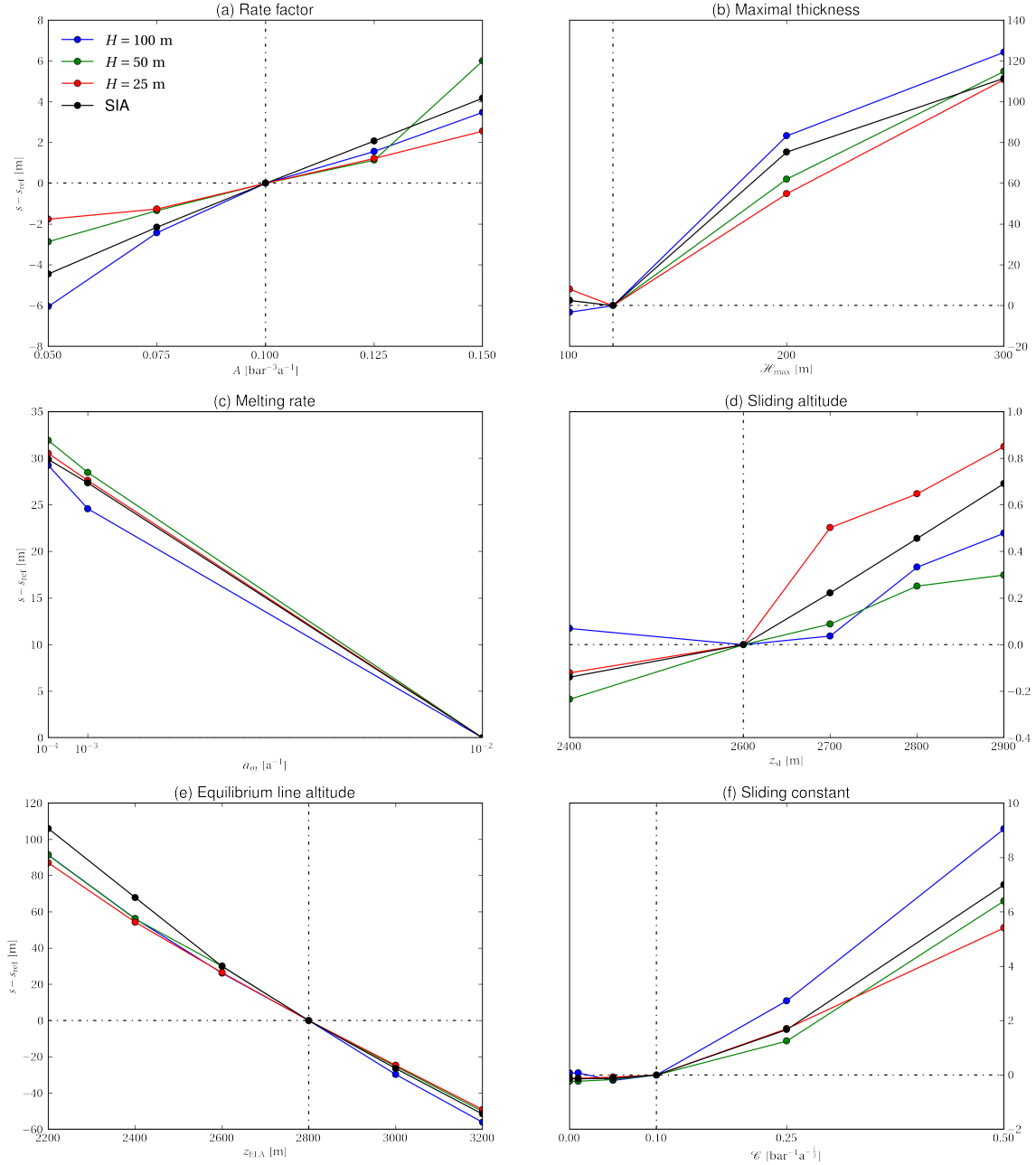


Figure 4.3 – Comparison of the surface topography discrepancy obtained down-glacier after 10 a-simulations with different values of both the Stokes and shallow ice model parameters. The black dash-dotted lines represent the reference values (4.1) in each case, while the solid coloured lines the discrepancies obtained with different mesh sizes and also in the SIA. The solid black line stands for the SIA estimation with grid spacing $\Delta x = 25$ m. Note that the cases $z_{\text{sl}} = 2400$ m and $\mathcal{C} = 0 \text{ bar}^{-1} \text{a}^{-\frac{1}{3}}$ correspond to non-sliding dynamics. No significant discrepancy can be noticed in these numerical experiments. Moreover, the SIA sensitivity is analogous to that of the Stokes flow.

As shown by Figure 4.2, the sensitivities of the selected parameters are different on the chosen domains of variation

that correspond to the ranges usually admitted in glaciology. The order of magnitude of these parameters is more or less known and the subplots of the Figure depict parameter variations in the admitted order of magnitude for each of the parameters. Figure 4.2 clearly evinces that the most significant changes in surface topography are induced by variations in ice thickness, equilibrium line altitude, melting rate, and sliding constant (Figures 4.2(b), (e), (c), and (f)). It is actually worth noticing that rate factor perturbations are responsible for non-negligible surface responses (Figure 4.2(a)), although of the very small order of magnitude $10^{-22} \text{ Pa}^{-3} \text{ s}^{-1}$, when the time frame is large enough. Without surprise, no large surface perturbation is observed after 1 a, save when the ice thickness is increased (Figure 4.2(b)). With increasing time frame, the surface mass-balance distribution and the sliding constant gain more and more influence.

In addition, it can be noticed that the final surface topography elevation gets higher than the reference one with lower melting rate or ELA, because snow accumulates more. Furthermore, since ice velocity is directly proportional to the rate factor, the whole dynamics is slower and ice accumulation and ablation are amplified for small A . For large A , the ice velocity is higher, and the ice flows faster from the top of the glacier to its tongue. Therefore, more ice is advected from the top to the bottom, whence a decrease in the surface elevation up-glacier and an increase down-glacier. Next, when the ice thickness is increased, the surface altitude gets lower up-glacier and higher down-glacier. Finally, the sliding altitude does not seem to play a fundamental role in a 10 a-simulation, while the sliding constant leaves the up-glacier surface unchanged and is responsible for a slight surface elevation increase down-glacier because more ice is transported. As with increasing A , an increasing \mathcal{C} involves larger ice velocities down-glacier, whence a surface topography of higher elevation in that region. To complement these observations, Figure 4.3 shows that the finite-element mesh size affects most significantly the results when the sliding parameters are varied, whatever the time frame is. Finally, the shallow ice model has sensitivities to the investigated parameters analogous to those of the Stokes model.

4.2 Influence of subglacial bumps on various geometries

In this second experiment, the idea is to observe how a basal perturbation is transferred up to the glacier's surface in both the Stokes and shallow ice approximations. Moreover, the influence of the glacier's geometry is investigated. To this end, for each glacier, we first generate three bedrock topographies different from the original one in that each of them is supplemented with one single downward, convex, 100 m-wide and deep bump at the locations indicated by the solid black crosses on the top left contour plot of Figures 4.4, 4.5, 4.6, and 4.7 representing the map domains of Silvretta, Gries, Muragl, and Rhône glaciers respectively. The profile plots **a**, **b**, and **c** in these Figures represent the results of the experiment when the Stokes (continuous lines) and shallow ice (dashed lines) approximations are applied. Note that, in the Stokes approximation, the perturbations are about 4 finite-elements in the (x, y) -plane and 16 along the z -axis. After that, focus is put on the middle of each of these glaciers where the bumps' amplitude is varied so as to see how the bedrock-to-surface perturbation transfer depends on it. All the results presented in this section refer to synthetical glacier evolutions over 10 a.

Overall, a few phenomena can easily be noticed. First, the extent of surface topography response depends on the glacier's geometry. For example, a particularly large extent is noticeable on Figures 4.4(b)-(c) and 4.6(a)-(b)-(c). The oscillations in the surface topography shown on Figure 4.6 are due to the numerical methods applied for the solving of the Stokes and VOF equations. In all cases, the downward bump on the bedrock topography results in a damped combination of a downward and an upward bump. On Rhône glacier, which is the deepest of the set, the only non-negligible surface response is detected up-glacier (Figure 4.7(a)). At the other locations, it can hardly be noticed, as it is very small compared to the local ice thickness. Furthermore, surface responses have about the same extent in both the Stokes and shallow ice approximations, although the response's amplitude is different. Indeed, the transferred amplitude is 2 to 5 times smaller in the SIA. As far as these numerical experiments show, the SIA

diffusion operator filters more information than its Stokes counterpart. This is not surprising, as the SIA surfaces are always smooth, whatever the bedrock's shape is, contrary to the Stokes output surfaces.

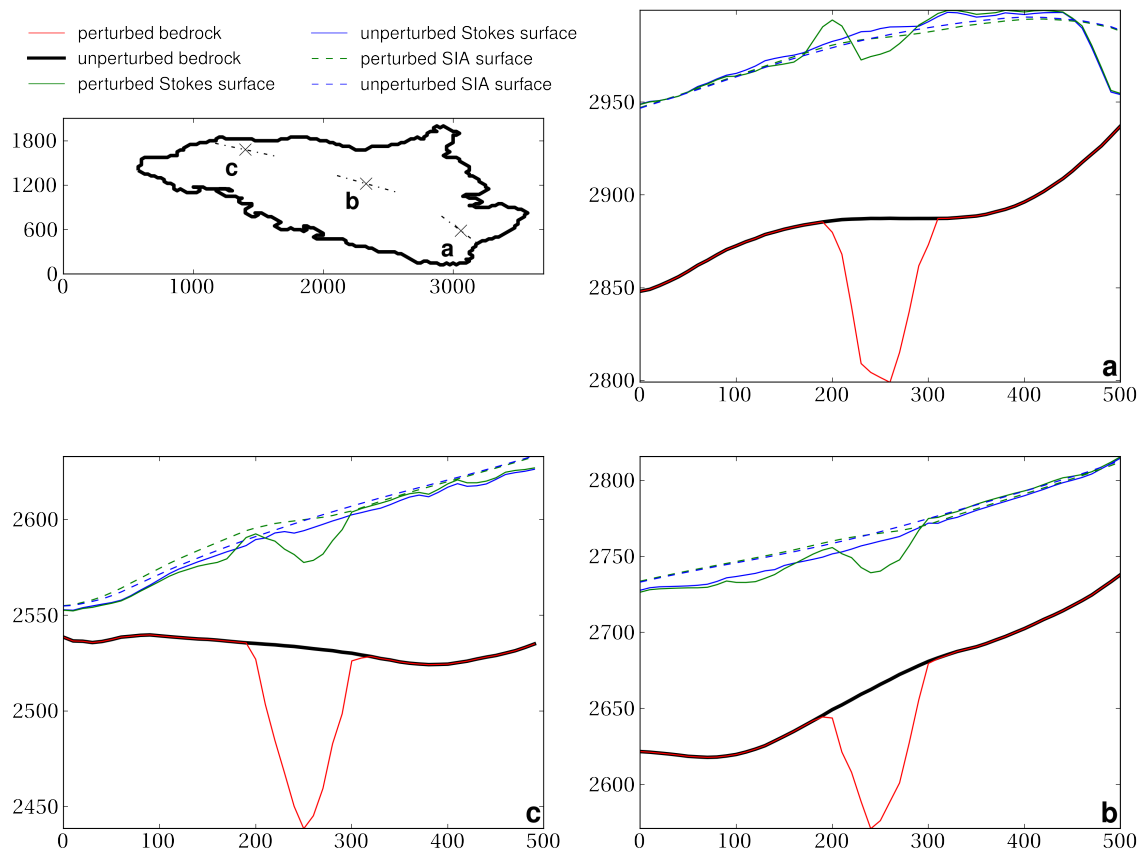


Figure 4.4 – Comparison of the surface topographies of Silvretta glacier obtained after an evolution of 10 a in the Stokes (continuous lines) and the shallow ice (dashed lines) approximations, starting from the same initial surface but perturbed bedrock topographies, to the reference surface topography (blue lines) obtained from a simulation initialized with the original bedrock topography. The perturbation consists in a downward, convex bump of deepness and width 100 m at the locations indicated by a cross in the contour plot. The continuous, thick, black line represents the original bedrock topography, the continuous, thin, red line the perturbed bedrock topography, the blue lines the unperturbed final surface topographies, and the green lines the final surface topographies resulting from the simulation with the perturbed bedrock topography. The dashed lines in the contour plot represent the lines along which the profiles are plotted. In each subplot, ice flows from the right to the left.

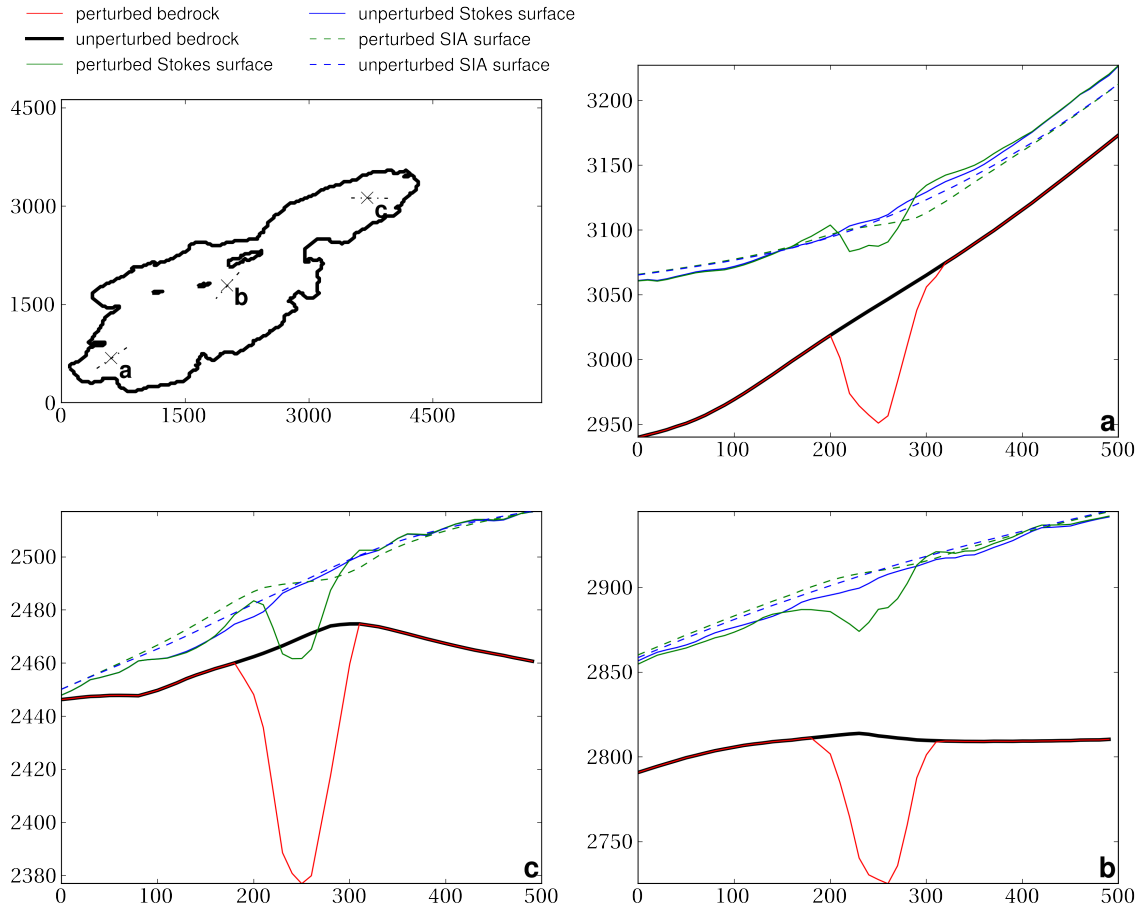


Figure 4.5 – Comparison of the surface topographies of Gries glacier obtained after an evolution of 10 a in the Stokes (continuous lines) and the shallow ice (dashed lines) approximations, starting from the same initial surface but perturbed bedrock topographies, to the reference surface topography (blue lines) obtained from a simulation initialized with the original bedrock topography. The perturbation consists in a downward, convex bump of deepness and width 100 m at the locations indicated by a cross in the contour plot. The continuous, thick, black line represents the original bedrock topography, the continuous, thin, red line the perturbed bedrock topography, the blue lines the unperturbed final surface topographies, and the green lines the final surface topographies resulting from the simulation with the perturbed bedrock topography. The dashed lines in the contour plot represent the lines along which the profiles are plotted. In each subplot, ice flows from the right to the left.

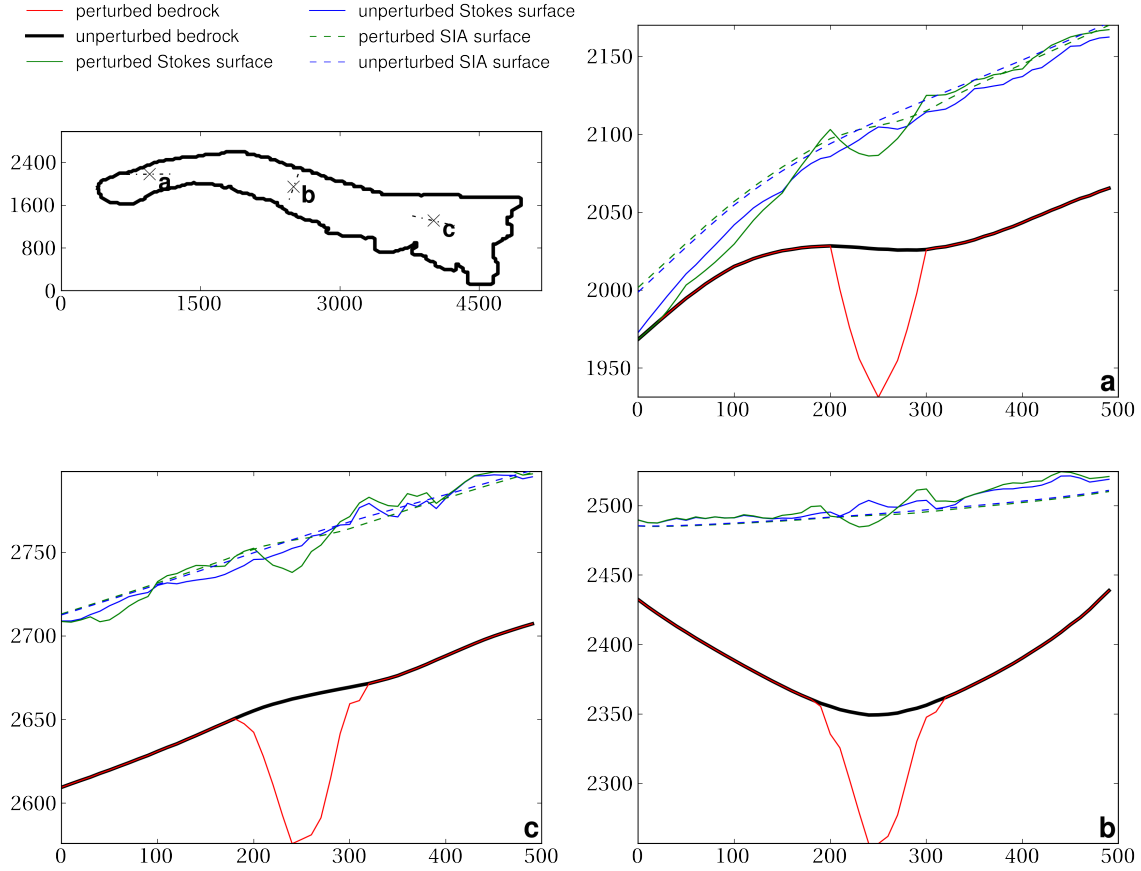


Figure 4.6 – Comparison of the surface topographies of Muragl glacier obtained after an evolution of 10 a in the Stokes (continuous lines) and the shallow ice (dashed lines) approximations, starting from the same initial surface but perturbed bedrock topographies, to the reference surface topography (blue lines) obtained from a simulation initialized with the original bedrock topography. The perturbation consists in a downward, convex bump of deepness and width 100 m at the locations indicated by a cross in the contour plot. The continuous, thick, black line represents the original bedrock topography, the continuous, thin, red line the perturbed bedrock topography, the blue lines the unperturbed final surface topographies, and the green lines the final surface topographies resulting from the simulation with the perturbed bedrock topography. The dashed lines in the contour plot represent the lines along which the profiles are plotted. In subplots (a) and (c), ice flows from the right to the left. In subplot (b), ice flows perpendicularly to the profile. The oscillatory behavior of the surface topography is due to mesh effects.

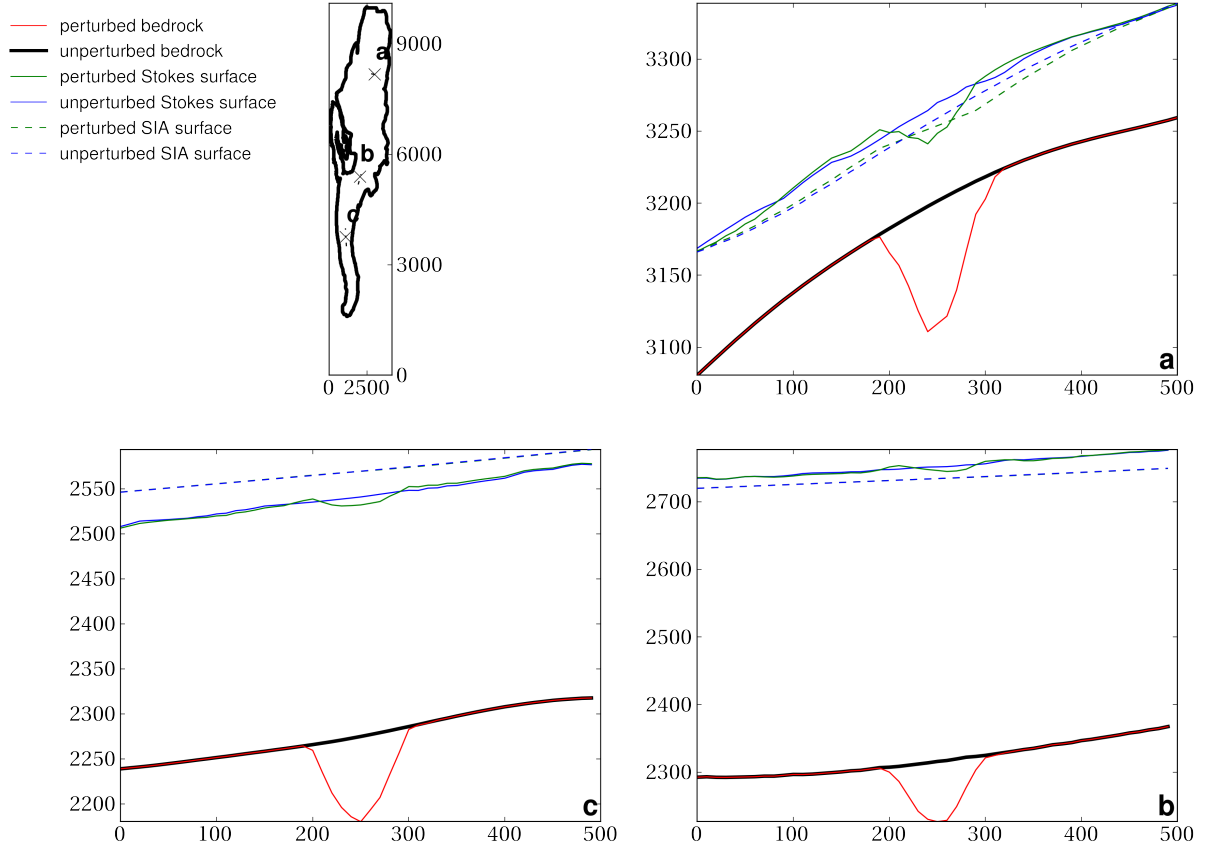


Figure 4.7 – Comparison of the surface topographies of Rhône glacier obtained after an evolution of 10 a in the Stokes (continuous lines) and the shallow ice (dashed lines) approximations, starting from the same initial surface but perturbed bedrock topographies, to the reference surface topography (blue lines) obtained from a simulation initialized with the original bedrock topography. The perturbation consists in a downward, convex bump of deepness and width 100 m at the locations indicated by a cross in the contour plot. The continuous, thick, black line represents the original bedrock topography, the continuous, thin, red line the perturbed bedrock topography, the blue lines the unperturbed final surface topographies, and the green lines the final surface topographies resulting from the simulation with the perturbed bedrock topography. The dashed lines in the contour plot represent the lines along which the profiles are plotted. In subplots (b) and (c), ice flows from the right to the left. In subplot (a), ice flows perpendicularly to the profile.

Figure 4.8 shows how the size of the bump located in the middle of the glacier influences the surface topography on the four different geometries. This is the bump located at position **b** on Figures 4.4-4.7 with varying amplitude and a width always equal to the amplitude. In both the Stokes and shallow ice approximations, the maximal response with the maximal bump size is comparable in magnitude. However, the way the response increases with the amplitude is different. In the shallow ice model, the increase seems to be parabolic, while it is much slower in the Stokes model. Furthermore, when subjected to small base perturbations, the shallow ice surface response is very small compared to the Stokes response. In the small amplitude region, the shallow ice filters more information than the Stokes approximation. In the large amplitude region, the tendency is inverted, as the Stokes response apparently grows more and more slowly.

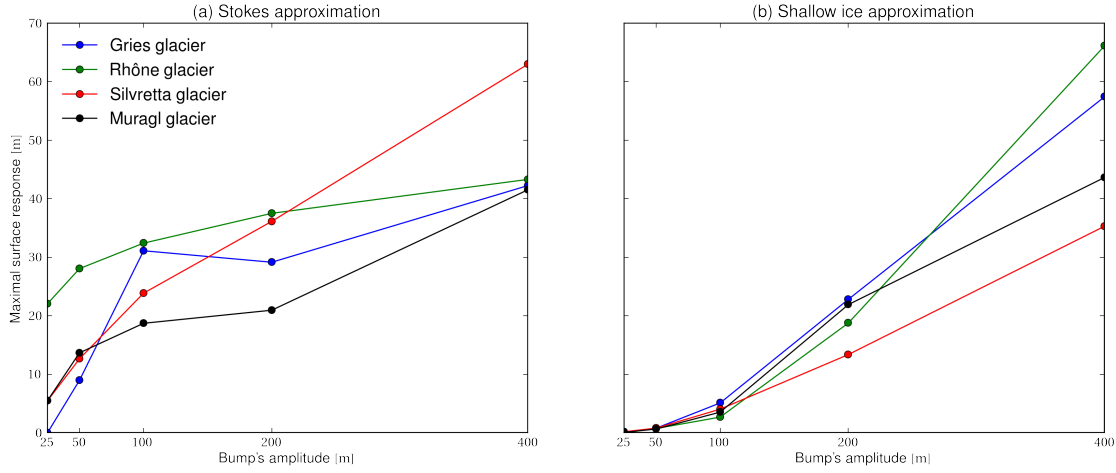


Figure 4.8 – Variation of the surface response as a function of the bedrock perturbation's amplitude, in both the Stokes (a) and the shallow ice (b) approximations, on Gries, Rhône, Silvretta, and Muragl glaciers. Each time, the bump's width is equal to its amplitude. The basal perturbation is performed in the middle of the glacier, corresponding to the middle plots in Figures 4.4-4.7. The surface response represented here is the maximal response detected over the whole glacier's surface, hence not necessarily right up the bump. All the Stokes simulations are performed on the finite-element mesh of size $H = 25$ m, except that on Rhône glacier, which is performed on a mesh of size $H = 50$ m because of computer resources. This explains the discrepancy with the results depicted on Figure 4.7, where the simulation was carried out on the finer mesh. The maximal surface response obtained is comparable in both approximations when the amplitude is large enough. For small amplitudes, the shallow ice transfers less information to the surface than the Stokes approximation. However, the increase in the SIA exhibits a parabolic behavior, contrary to the other, where the transfer increases much less with the bump's amplitude.

4.3 Conclusion

This chapter provides a useful sensitivity analysis of the Stokes and shallow ice flow models. It first describes how the surface topography responds to variations in the model parameters on the fixed geometry of Silvretta glacier. Numerical experiments over 1 and 10 a are discussed. Clearly, the larger the time frame, the larger the surface topography response. Hence, the analysis is essentially based on the results obtained on the 10 a-simulations. The example geometry of Silvretta glacier is not very sensitive to the sliding altitude z_{sl} . However, the other parameters influence the glacier's evolution significantly. The equilibrium line altitude and the glacier's thickness are responsible for the most important changes in the surface topography at final time, whose orders of magnitude are about 16 times the finite element size in the vertical direction.

Then, the influence of a bump positioned at several locations in the bedrock topography is studied on several alpine glaciers. With the typical ice thickness of the considered geometries, the surface response decreases in relative value. The shallow ice flow filters more subglacial information than the Stokes flow. With the perturbation's amplitude and width, the surface response increases parabolically in the shallow ice approximation, while it slowly flattens in the Stokes approximation.

The bedrock-to-surface perturbation transfer is not localized, but spreads around the perturbation's neighborhood instead. Moreover, different bedrock topographies can provide more or less the same final surface topographies, with discrepancies of the order of the finite element size. All the examples supplied in this chapter illustrate this phenomenon. Each time, three different bedrocks are inputted to the two models. Everytime, the surface response is comparable in amplitude to the mesh size, although the bedrock topography is augmented with a large bump compared to both the glaciers' typical thickness and mesh size. The problem of inferring the subglacial topography

from surface topography data is ill-posed and therefore very difficult to solve, which partly explains the sparsity of research results on the topic.

The inaccuracies in the ice rheology, surface mass-balance, and sliding law model parameters add up even more complications to the intrinsic ill-posedness of the inverse problem. Indeed, not only the bedrock topography is unknown but also most of the model parameters. In the end of this thesis, a shape optimization algorithm is tailored that computes some of these parameters and the bedrock topography simultaneously on transient shallow ice glaciers.

The next chapter exposes various methods that do not rely on optimal control techniques. First, the stationary inverse method is introduced, before its quasi-stationary extension, both in the shallow ice approximation only. Then, the transient inverse method is presented, that makes use of any of the shallow ice or Stokes approximations.

5 Transient Inverse Method

Usually, simplified, low order models (*e.g.* those shown in Table 2.2) are used for the determination of a glacier's subglacial topography, because the ice thickness is often linked directly to some quantity known from measurements. For example, the methods based on the perfect plasticity assumption use the proportional relationship between the thickness and the constant basal shear stress [90, 54, 93, 75]. The method presented in [34] uses the relation between the ice volume flux and the thickness in the parallel-sided slab approximation. In this context, two space dimensions and stationarity in time are often assumed. The procedures are then applied to selected glacier flow lines, before an interpolation of the scattered data is performed in order to obtain a complete three-dimensional ice thickness distribution over the whole glacier's map domain.

This chapter shows several new ways to invert the shallow ice equations. Instead of explicitly computing the ice thickness distribution within the aforementioned assumptions, implicit equations are solved iteratively for the ice thickness, in order to minimize the surface topography misfit at final time without the use of any gradient information. The procedures exposed below are improved extensions of existing methods in stationary fluid dynamics [43, 58]. They are developed in the two and three-dimensional shallow ice as well as in the three-dimensional Stokes ice approximations. A gradient-based method is proposed in the next chapter.

The chapter's outline is the following: First, the method developed in [34] is recalled and slightly reformulated, since the main motivation behind this thesis is to make it executable in a more automatic and efficient way. Second, the stationary inverse method is introduced, that infers the ice thickness distribution from a glacier's steady surface topography and is the basis of the quasi-stationary inverse method. This method is inspired from an algorithm that computes the bed of steady rivers from surface observations [43]. It is formulated in the two-dimensional, shallow ice approximation, without consideration of sliding, and with a time-independent surface mass-balance. Third, the quasi-stationary inverse method is described, which is basically the stationary inverse method where the source term is replaced with the so-called apparent surface mass-balance, *i.e.* the balance between the surface mass-balance and the ice thickness variation. It is extended to three-space dimensions in the shallow ice approximation and takes sliding and a time-dependent surface mass-balance into account. Next, the transient inverse method is formulated, which is a procedure that iteratively updates the bedrock topography with the surface topography misfit and uses the true surface mass-balance. Both its shallow and Stokes ice descriptions are presented. Then, numerical results are exposed, most of which are two-dimensional, as the computations are much cheaper, before a short sensitivity analysis. The methods are finally applied to real-world data and conclusions are drawn.

5.1 Parallel-sided slab estimation

A simple inversion method can be tailored based on the two-dimensional, parallel-sided slab approximation (PSSA) of ice flow [34], which is a special case of the two-dimensional shallow ice approximation where the derivative of the surface topography with respect to the position is replaced with the tangent of the surface angle. The two-dimensional, shallow ice velocity reads, looking back at equation (2.13), without the sliding contribution,

$$u_x(x, z) = -\frac{n+2}{n+1} \Gamma \left(\mathcal{H}^{n+1} - (s(x) - z)^{n+1} \right) |s'(x)|^{n-1} s'(x), \quad (5.1)$$

where $s'(x)$ is the derivative of s with respect to x .

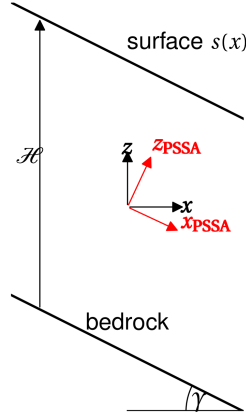


Figure 5.1 – Two-dimensional parallel-sided slab approximation of a glacier. The surface topography is parallel to the bedrock topography and has an inclination angle γ . The constant ice thickness is expressed in the axes that follow the ice flow whose direction is given by axis x_{PSSA} .

In the PSSA, the bedrock and surface topographies are assumed parallel with constant inclination angle $0 < \gamma < \frac{\pi}{2}$ everywhere (see Figure 5.1). Both topographies are flat in the $(x_{\text{PSSA}}, z_{\text{PSSA}})$ -system of axes. Therefore,

$$s'(x) = -\tan(\gamma) \quad (5.2)$$

and the ice velocity becomes

$$u_x(x, z) = \frac{n+2}{n+1} \Gamma \left(\mathcal{H}^{n+1} - (s(x) - z)^{n+1} \right) \tan^n(\gamma) = \frac{n+2}{n+1} \Gamma \sin^n(\gamma) \left(\mathcal{H}_{\text{PSSA}}^{n+1} - \left(\mathcal{H}_{\text{PSSA}} - z_{\text{PSSA}} \right)^{n+1} \right) \cos(\gamma),$$

where

$$\mathcal{H}_{\text{PSSA}} = \frac{\mathcal{H}}{\cos(\gamma)}. \quad (5.3)$$

Hence, the PSSA ice velocity in the flow direction x_{PSSA} is

$$u^{\text{PSSA}} = \frac{n+2}{n+1} \Gamma \sin^n(\gamma) \left(\mathcal{H}_{\text{PSSA}}^{n+1} - \left(\mathcal{H}_{\text{PSSA}} - z_{\text{PSSA}} \right)^{n+1} \right). \quad (5.4)$$

From this expression, the ice thickness can be easily extracted. Indeed, the integration of the velocity from the base

to the surface yields a volume flux

$$Q^{\text{PSSA}} = \int_0^{\mathcal{H}_{\text{PSSA}}} u^{\text{PSSA}}(z) dz = \Gamma \sin^n(\gamma) \mathcal{H}_{\text{PSSA}}^{n+2}, \quad (5.5)$$

whence the relation used in [34], without sliding on the base:

$$\mathcal{H}_{\text{PSSA}} = \sqrt[n+2]{\frac{Q^{\text{PSSA}}}{\Gamma \sin^n(\gamma)}}. \quad (5.6)$$

In this approximation, sliding can be taken into account by multiplying the denominator under the root with a constant that is tuned to field data. The smaller the constant, the more important the sliding contribution. The effect of such a small constant is then to increase the ice thickness where sliding is assumed. In the shallow ice approximation, the incorporation of sliding can be improved as suggested in Section 5.3.

Estimates of the volume flux Q^{PSSA} are obtained from an apparent surface mass-balance taking into account the observed changes in the ice thickness over some time frame. Then, formula (5.6) is applied to the principal flow line of the glacier's ice catchments that are approximate parallel-sided slabs of the glacier where the surface angle does not vary much. Finally, from these catchments, their outlines, and the glacier's outline, an ice thickness distribution can be calculated. In order to prevent divergence from happening because of possibly small angles γ , the glacier's surface topography is filtered with a lower slope limit. The method is summarized in Algorithm 3.

Algorithm 3 PSSA Inverse Method [34]

identify the glacier's outline
identify the glacier's principal flow lines
identify the glacier ice catchments and their outlines corresponding to each flow line
determine Q^{PSSA} (e.g. from measurements)
for each catchment **do**
 compute the ice thickness with equation (5.6) with a lower slope limit
end for
smooth the obtained distribution with a Gaussian filter of constant extension

In the PSSA, the basal shear stress τ and the ice thickness are linked together through the relation [48, 76]

$$\mathcal{H}_{\text{PSSA}}(x) = \frac{\tau}{f \rho g \sin(\gamma(x))}, \quad (5.7)$$

where f is a shape factor. Under the perfect plasticity assumption, the shear stress τ is assumed constant. In the most recently developed inversion method based on perfectly plastic ice, it is approximated with the formula [76]

$$\tau = 0.005 + 1.598\Delta H - 0.435\Delta H^2, \quad (5.8)$$

where ΔH is the elevation range of the glacier, and the three numerical values were obtained after tuning on a large sample of glaciers. The idea is simple, but the resulting estimation is crude compared to the SIA. To illustrate this point, let us assume that a bedrock and a steady surface topography as well as a surface mass-balance are available that satisfy equation (2.32) for the two-dimensional profile line of Muragl glacier depicted by Figure 5.2. Knowing which bedrock topography, namely the black line labeled with “exact bedrock” on Figure 5.2, makes the glacier's surface evolve towards the stationary surface delineated in black and labeled with “stationary surface”, the bedrock computed by means of equation (5.7), delineated by the green curve on Figure 5.2, is compared to the exact bedrock topography. The green curve actually represents the best fit to the profile of the solution bedrock topography, with

$\Delta H = 400$ m and a shape factor $f = 0.5$, which are not even the advised values. A significant large defect in the estimation can clearly be noticed up-glacier, while the estimation of the tongue's thickness is fine. Furthermore, estimation (5.7)-(5.8) cannot take into account the measured evolution of transient glaciers.

Figure 5.2 also depicts the estimation performed by the integrated shallow ice equation, that corresponds to (5.10) presented in the next section. Clearly, the integrated shallow ice estimation is more precise (in red), the largest errors being at the glacier's boundaries, because the derivative of the surface topography takes very small values at these locations. As shows Figure 5.2(b), this drawback is corrected by the finite-difference method presented in the next sections.

Algorithm 3 requires lots of pre-processing work that cannot be automated in an easy way. Moreover, it is based on a simplified treatment of sliding. The purpose of the next sections is to design more practical and efficient ways to compute the ice thickness distribution along with a more physical incorporation of the glacier's sliding.

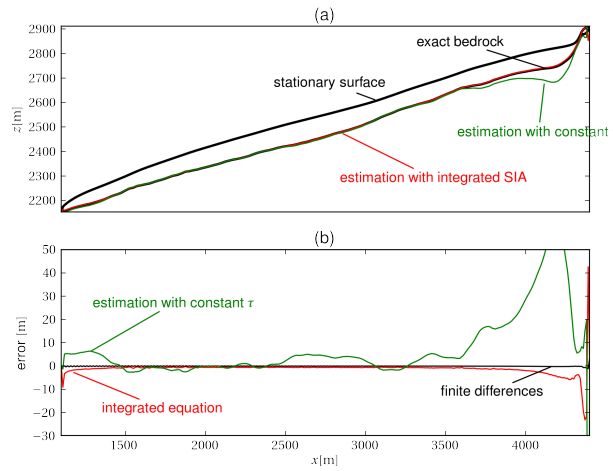


Figure 5.2 – (a) Application of integral formula (5.10) to a profile line of Muragl glacier whose surface gradient is never vanishing. The estimation is good, except at the boundaries, as shown in (b), which compares the maximal error in the thickness estimated from integral (5.10), estimation (5.7) and the iterative method presented below (indicated with the label “finite differences”).

5.2 Stationary Inverse Method

The inversion approach exposed in this section for a steady-state surface geometry is inspired by an existing method used to reconstruct the bed topography of a steady river based on the Saint-Venant shallow water approximation [43]. Now, the purpose is to develop an algorithm that reconstructs the subglacial topography of steady glaciers. In the next section, transient glaciers are dealt with the quasi-stationary inverse method which is its direct extension. While this section is exclusively devoted to a two-dimensional algorithm, the next section extends all the results to three space dimensions, in the shallow ice approximation.

Consider the steady-state problem corresponding to (2.32) in the case where the stationary surface s is known and no sliding applies. The problem is now: *given the stationary surface s and surface mass-balance \mathcal{B} , find the ice thickness $\mathcal{H} : [x_1, x_2] \rightarrow \mathbb{R}$ such that*

$$\begin{cases} \Gamma(\mathcal{H}^{n+2}|s'|^{n-1}s')' = -\mathcal{B}, & x_1 < x < x_2, \\ \mathcal{H} \geq 0, & x_1 < x < x_2, \\ \mathcal{H} = 0, & x = x_1, x_2, \end{cases} \quad (5.9)$$

where s' stands for the derivative of s with respect to x and the surface mass-balance \mathcal{B} is applied only at the locations where $\mathcal{H} > 0$. Once \mathcal{H} is known, the bedrock topography can be deduced from the relation $b = s - \mathcal{H}$. This problem can be solved by direct integration whenever $s'(x) \neq 0$:

$$\mathcal{H}(x) = \left(- \frac{\int_{x_1}^x \mathcal{B}(\xi) d\xi}{\Gamma |s'(x)|^{n-1} s'(x)} \right)^{\frac{1}{n+2}}. \quad (5.10)$$

Note that the above formula yields $\mathcal{H}(x_2) = 0$ provided that either

$$\int_{x_1}^{x_2} \mathcal{B}(x) dx = 0 \quad (5.11)$$

or

$$\lim_{x \rightarrow x_2} s'(x) = \pm\infty \quad \text{and} \quad \int_{x_1}^{x_2} \mathcal{B}(x) dx < \infty. \quad (5.12)$$

Formula (5.10) corresponds to the PSSA estimation (5.6) where

$$Q^{\text{PSSA}} = - \int_{x_1}^x \mathcal{B}(\xi) d\xi, \quad (5.13)$$

which is exactly (5.9)₁, since the ice volume flux (2.30) specializes to

$$Q = \Gamma \mathcal{H}^{n+2} |s'|^{n-1} s' \quad (5.14)$$

in the two-dimensional, stationary shallow ice approximation without sliding.

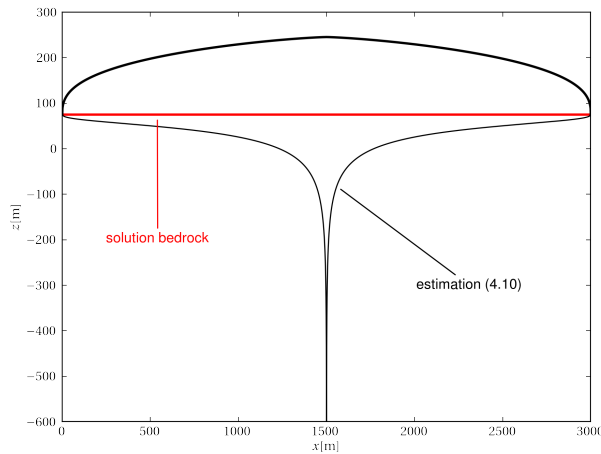


Figure 5.3 – Ice thickness estimation with equation (5.10) in the case of an ice cap. Note the divergence in the middle of the glacier where the surface slope goes to zero.

When the glacier's surface topography is flat, *i.e.* $s'(x) = 0$, the ice thickness cannot be computed with (5.10). For example, flat ice caps cannot be inverted with this relation, as depicted by Figure 5.3. In fact, formula (5.10) suffers from a few drawbacks. First, the divergence where $s'(x) = 0$ is brought by the space integration and is not an intrinsic property of problem (5.9). Second, estimation (5.10) is very unstable when the input data are not regular enough. In addition, an extension of this formula to sliding glaciers can only be done through a further approximation

of the actual SIA. Finally, and most importantly, this analytical expression cannot be generalized to three space dimensions because, in this case, equation (5.9) becomes a divergence equation [12, 13].

Instead of regularizing equation (5.10) by filtering the surface topography with a lower slope limit (*e.g.* [34, 75, 76]), we propose to solve the “pseudo-time-dependent” problem corresponding to (5.9), namely

$$\begin{cases} \frac{\partial \mathcal{H}}{\partial t} - \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial s}{\partial x} \right) = \mathcal{B}, & x_1 < x < x_2, \quad t > 0, \\ \mathcal{H} = \mathcal{H}_0(x), & x_1 < x < x_2, \\ \mathcal{H} \geq 0, & x_1 < x < x_2, \quad t > 0, \\ \mathcal{H} = 0, & x = x_1, x_2, \quad t \geq 0, \end{cases} \quad (5.15)$$

with finite differences, where t is introduced to help solving (5.9), \mathcal{H}_0 is an initial guess, s a fixed surface, and \mathcal{D}_f is given by (2.33). Given the stationary surface topography s , the aim is to find a solution \mathcal{H} so that $\frac{\partial \mathcal{H}}{\partial t} \rightarrow 0$ as $t \rightarrow \infty$, which defines the subglacial topography with $b = s - \mathcal{H}$. Note that this equation is exactly the same as (2.32), with the difference that here the surface topography s is known. In the forward formulation (2.32), the latter topography is sought. When the surface topography s is known, the originally elliptic problem (2.33) becomes hyperbolic. This non-linear transport problem has the particularity of having a non-zero source and Dirichlet boundary conditions. To solve (5.15), focus is put on the implicit version of scheme (3.32), namely

$$\frac{\mathcal{H}_j^{\ell+1} - \mathcal{H}_j^\ell}{\Delta t} = \frac{1}{2\Delta x} \left[\left((\mathcal{D}_f)_{j+\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j+\frac{1}{2}}^{\ell+1} \right) \frac{s_{j+1} - s_j}{\Delta x} - \left((\mathcal{D}_f)_{j-\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j-\frac{1}{2}}^{\ell+1} \right) \frac{s_j - s_{j-1}}{\Delta x} \right] + \mathcal{B}_j^\ell, \quad 1 \leq j \leq N_x, \quad 0 \leq \ell \leq M, \quad (5.16)$$

where the surface components s_j are fixed to the values corresponding to the stationary surface topography. In most cases, Newton’s method to solve the inverse problem (5.15) with scheme (5.16) does not converge. This is because the inverse equation is no longer diffusive. It is a stationary hyperbolic equation that is known to be difficult to solve [31, 85]. Therefore, we propose to add to the equation a diffusion term that has more weight when the variations in thickness are the largest and is close to the diffusion term of the forward model [73, 85]:

$$-\varepsilon \frac{\partial}{\partial x} \left(\tilde{\mathcal{D}}_f \frac{\partial \mathcal{H}}{\partial x} \right) + \frac{\partial \mathcal{H}}{\partial t} = \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial s}{\partial x} \right) + \mathcal{B}, \quad (5.17)$$

where $\varepsilon > 0$ must be as small as possible and, looking back at (2.33),

$$\tilde{\mathcal{D}}_f = \Gamma \mathcal{H}^{n+2} \left| \frac{\partial \mathcal{H}}{\partial x} \right|^{n-1}. \quad (5.18)$$

To solve (5.17), we update the original scheme to

$$\begin{aligned} \frac{\mathcal{H}_j^{\ell+1} - \mathcal{H}_j^\ell}{\Delta t} = & \frac{1}{2\Delta x} \left[\left((\mathcal{D}_f)_{j+\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j+\frac{1}{2}}^{\ell+1} \right) \frac{s_{j+1} - s_j}{\Delta x} - \left((\mathcal{D}_f)_{j-\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j-\frac{1}{2}}^{\ell+1} \right) \frac{s_j - s_{j-1}}{\Delta x} \right] \\ & + \frac{\varepsilon}{2\Delta x} \left[\left((\tilde{\mathcal{D}}_f)_{j+\frac{1}{2}}^\ell + (\tilde{\mathcal{D}}_f)_{j+\frac{1}{2}}^{\ell+1} \right) \frac{\mathcal{H}_{j+1}^{\ell+1} - \mathcal{H}_j^{\ell+1}}{\Delta x} - \left((\tilde{\mathcal{D}}_f)_{j-\frac{1}{2}}^\ell + (\tilde{\mathcal{D}}_f)_{j-\frac{1}{2}}^{\ell+1} \right) \frac{\mathcal{H}_j^{\ell+1} - \mathcal{H}_{j-1}^{\ell+1}}{\Delta x} \right] + \mathcal{B}_j, \end{aligned} \quad (5.19)$$

with the same boundary conditions as in the forward problem and where

$$(\tilde{\mathcal{D}}_f)_{j-\frac{1}{2}}^\ell = \Gamma \left(\frac{\mathcal{H}_j^\ell + \mathcal{H}_{j-1}^\ell}{2} \right)^{n+2} \left| \frac{\mathcal{H}_j^\ell - \mathcal{H}_{j-1}^\ell}{\Delta x} \right|^{n-1}. \quad (5.20)$$

Convergence is reached when the stopping criterion

$$\|\mathcal{H}^{\ell+1} - \mathcal{H}^{\ell}\| = \max_{1 \leq j \leq N} |\mathcal{H}_j^{\ell+1} - \mathcal{H}_j^{\ell}| < \delta \quad (5.21)$$

is fulfilled for a real constant $\delta > 0$. Note that this regularization modifies the original model, so that the solution of (5.17) is not expected to match the solution bedrock b^* exactly.

5.3 Quasi-stationary Inverse Method

The stationary inverse method introduced in the previous section is not very useful as it is in practical applications, because generally, a glacier is not stationary. Hence, in this section, its extension to transient glaciers in two space dimensions is presented. Then, its three-dimensional generalization that takes into account the time-dependence of the surface mass-balance and the sliding law is explained in the shallow ice approximation. A complete analysis of the method is performed in two dimensions of space. Its features are essentially the same in three space dimensions, hence only a short numerical validation is presented in this case. An extension of the algorithm exposed here to Stokes ice flows is possible but complicated, hence it is not developed in this thesis.

5.3.1 Two-dimensional formulation

Consider the transient problem corresponding to (2.32) in the case where the initial and final surfaces and the surface mass-balance \mathcal{B} are known. The initial surface is denoted by s_i and the exact final surface is s^{obs} . The aim is to minimize the discrepancy between the known final surface s^{obs} and the computed final surface $s_f(x) = b(x) + \mathcal{H}(x, t_f)$ with equation (2.32). The inversion problem has two unknowns, namely the surface geometry s , which is only known at times t_i and t_f , and the bedrock topography b (or, equivalently, the ice thickness \mathcal{H}). The time-dependent inverse problem can be stated as follows: *given the glacier surfaces s_i at time t_i and s^{obs} at time t_f as well as the surface mass-balance time series \mathcal{B} in the time interval $[t_i, t_f]$, find the bedrock geometry $b : [x_1, x_2] \rightarrow \mathbb{R}$ such that*

$$s^{\text{obs}}(x) = b(x) + \mathcal{H}(x, t_f), \quad x_1 < x < x_2, \quad (5.22)$$

under the constraint that \mathcal{H} satisfies the ice thickness equation

$$\left\{ \begin{array}{ll} \frac{\partial \mathcal{H}}{\partial t} = \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial(b + \mathcal{H})}{\partial x} \right) + \mathcal{B}, & x_1 < x < x_2, \quad t_i < t < t_f, \\ \mathcal{H} = s_i - b, & x_1 < x < x_2, \quad t = t_i, \\ \mathcal{H} \geq 0, & x_1 < x < x_2, \quad t_i \leq t \leq t_f, \\ \mathcal{H} = 0, & x = x_1, x_2, \quad t_i \leq t \leq t_f. \end{array} \right. \quad (5.23)$$

In other words, this means that a bedrock topography b is sought such that the glacier evolves from surface s_i to s^{obs} over the time interval $[t_i, t_f]$ under the influence of the surface mass-balance \mathcal{B} . Let us approximate $\frac{\partial \mathcal{H}}{\partial t}$ by $\frac{s^{\text{obs}} - s_i}{t_f - t_i}$, so that equation (5.23)₁ becomes

$$\frac{s^{\text{obs}} - s_i}{t_f - t_i} = \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial(b + \mathcal{H})}{\partial x} \right) + \mathcal{B}, \quad (5.24)$$

and define the so-called “apparent surface mass-balance” [34] with

$$\tilde{\mathcal{B}} = \mathcal{B} - \frac{s^{\text{obs}} - s_i}{t_f - t_i}. \quad (5.25)$$

Then, the quasi-stationary formulation of the transient inverse problem in the SIA is: *given the glacier surface topographies s_i and s^{obs} as well as the surface mass-balance time series \mathcal{B} in the time interval $[t_i, t_f]$, find the thickness $\mathcal{H} : [x_1, x_2] \rightarrow \mathbb{R}$ such that*

$$\begin{cases} \Gamma(\mathcal{H}^{n+2}|s'|^{n-1}s')' = -\tilde{\mathcal{B}}, & x_1 < x < x_2, \\ \mathcal{H} \geq 0, & x_1 < x < x_2, \\ \mathcal{H} = 0, & x = x_1, x_2, \end{cases} \quad (5.26)$$

where s can be any linear combination of s_i and s^{obs} . In our computations, we chose $s = s^{\text{obs}}$. When the thickness \mathcal{H} is known, then the bedrock topography can be deduced as before with the relation $b = s - \mathcal{H}$. Note that equation (5.26) is precisely (5.9), with a modified source term, that takes into account the glacier evolution over the desired time interval. In order to solve (5.26) numerically, we proceed as in the previous section and use the regularized scheme (5.19).

5.3.2 Three-dimensional formulation

In three space dimensions, with the sliding contribution included, equation (5.26) becomes the divergence equation

$$\begin{cases} \nabla \cdot (\mathcal{D} \nabla s) = -\tilde{\mathcal{B}}, & \text{in } \Omega_{\perp} \\ \mathcal{H} \geq 0, & \text{in } \Omega_{\perp} \\ \mathcal{H} = 0, & \text{on } \partial\Omega_{\perp}, \end{cases} \quad (5.27)$$

which is the quasi-stationary specialization of (2.31), where now the diffusivity is given by (2.20) and (2.21) instead of the two-dimensional, non-sliding version (2.33) of the previous sections. To solve this divergence equation, the same procedure as before applies, namely the following pseudo-time-dependent problem is considered:

$$\begin{cases} -\varepsilon \nabla \cdot (\tilde{\mathcal{D}} \nabla \mathcal{H}) + \frac{\partial \mathcal{H}}{\partial t} = \nabla \cdot (\mathcal{D} \nabla s) + \tilde{\mathcal{B}}, & \text{in } \Omega_{\perp}, \quad t > 0 \\ \mathcal{H} \geq 0, & \text{in } \Omega_{\perp}, \quad t > 0 \\ \mathcal{H} = 0, & \text{on } \partial\Omega_{\perp}, \quad t > 0, \end{cases} \quad (5.28)$$

where such a thickness \mathcal{H} is sought that satisfies $\frac{\partial \mathcal{H}}{\partial t} \rightarrow 0$ as $t \rightarrow \infty$, the surface topography s is fixed as before to $s = s^{\text{obs}}$, and

$$\tilde{\mathcal{D}} = \left(\Gamma \mathcal{H} + \Gamma_s \right) \mathcal{H}^{n+1} \left\| \nabla \mathcal{H} \right\|^{n-1}. \quad (5.29)$$

This extends problem (5.17) to three space dimensions and can be solved with the numerical scheme that generalizes (5.19). However, at the expense of a higher CPU time due to its worse stability, the following semi-implicit scheme is advocated, to keep the method as simple as possible:

$$\begin{aligned} \frac{\mathcal{H}_{jk}^{\ell+1} - \mathcal{H}_{jk}^{\ell}}{\Delta t} &= \frac{1}{\Delta x} \left[\mathcal{D}_{j+\frac{1}{2},k}^{\ell} \frac{s_{j+1,k} - s_{jk}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^{\ell} \frac{s_{jk} - s_{j-1,k}}{\Delta x} \right] + \frac{1}{\Delta x} \left[\mathcal{D}_{j,k+\frac{1}{2}}^{\ell} \frac{s_{j,k+1} - s_{jk}}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^{\ell} \frac{s_{jk} - s_{j,k-1}}{\Delta x} \right] \\ &+ \frac{\varepsilon}{\Delta x} \left[\tilde{\mathcal{D}}_{j+\frac{1}{2},k}^{\ell} \frac{\mathcal{H}_{j+1,k}^{\ell+1} - \mathcal{H}_{jk}^{\ell+1}}{\Delta x} - \tilde{\mathcal{D}}_{j-\frac{1}{2},k}^{\ell} \frac{\mathcal{H}_{jk}^{\ell+1} - \mathcal{H}_{j-1,k}^{\ell+1}}{\Delta x} \right] + \frac{\varepsilon}{\Delta x} \left[\tilde{\mathcal{D}}_{j,k+\frac{1}{2}}^{\ell} \frac{\mathcal{H}_{j,k+1}^{\ell+1} - \mathcal{H}_{jk}^{\ell+1}}{\Delta x} - \tilde{\mathcal{D}}_{j,k-\frac{1}{2}}^{\ell} \frac{\mathcal{H}_{jk}^{\ell+1} - \mathcal{H}_{j,k-1}^{\ell+1}}{\Delta x} \right] \end{aligned}$$

$$+ \tilde{\mathcal{B}}_{jk}, \quad (5.30)$$

where

$$\tilde{\mathcal{D}}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \left(\Gamma \mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + (\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} \right) \left(\mathcal{H}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell \right)^{n+1} \left(\tilde{\alpha}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell \right)^{n-1} \quad (5.31)$$

$$\tilde{\alpha}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = \left[\left(\frac{\mathcal{H}_{jk}^\ell - \mathcal{H}_{j-1,k}^\ell + \mathcal{H}_{j,k-1}^\ell - \mathcal{H}_{j-1,k-1}^\ell}{2\Delta x} \right)^2 + \left(\frac{\mathcal{H}_{jk}^\ell - \mathcal{H}_{j,k-1}^\ell + \mathcal{H}_{j-1,k}^\ell - \mathcal{H}_{j-1,k-1}^\ell}{2\Delta x} \right)^2 \right]^{\frac{1}{2}}. \quad (5.32)$$

Scheme (5.30) is solved for every couple of indices (j, k) such that the ice thickness is known to be larger than 0. Such a domain of indices can be deduced in practice from measurements, since one can clearly see what the glacier's outline is.

If not semi-implicit in time, this method can become complicated to implement. However, the stability condition is more restrictive than in the implicit discretization, hence increasing the computational time. Note that it is also possible to apply the algorithm consisting in the successive solving of (3.14), (3.15), (3.16), and (3.18), in both the semi-implicit and implicit time discretizations, with the addition of the regularization term. The results are similar but the method requires more coding. Such a complexity is not required here because a stationary state is sought.

Algorithm 3 can be simplified to Algorithm 4 which summarizes the quasi-stationary inverse method. With this method, there is no need any more to determine the flow lines by hand and then define the glacier's ice catchments. No additional algorithm is needed for the computation of Q^{PSSA} , since it is directly deduced from \mathcal{B} and $\frac{\partial \mathcal{H}}{\partial t}$. Moreover, sliding is taken into account more precisely, *i.e.* not as a constant in the denominator inside the root of equation (5.6). Finally, bedrock topography measurements can be easily included in this algorithm by replacing \mathcal{H}_{jk}^ℓ with its measured value in scheme (5.30) or by applying an extension of the method described in [14].

Algorithm 4 Quasi-stationary Inverse Method

initialize the glacier's ice thickness distribution
solve problem (5.28) with scheme (5.30)

To conclude this section, let us mention that the quasi-stationary inverse method can also be applied to the Stokes model. Indeed, the bedrock topography can be updated with the VOF function (2.26) instead of the surface topography. However, it is not trivial to develop a regularized version of the VOF equation (2.27). Our few numerical experiments to test such a procedure showed that there was no hope to converge without such a regularization. This idea is not developed further.

5.4 Transient Inverse Method

The numerical experiments in two space dimensions of Section 5.5.2 confirm that the quasi-stationary approximation gives worse estimations of the thickness when the time interval $[t_i, t_f]$ gets wider. This is due to the assumption of a uniform variation of the ice thickness in time and is undesirable because an approximation of at least equivalent quality is expected with the knowledge of the glacier's evolution over a longer time.

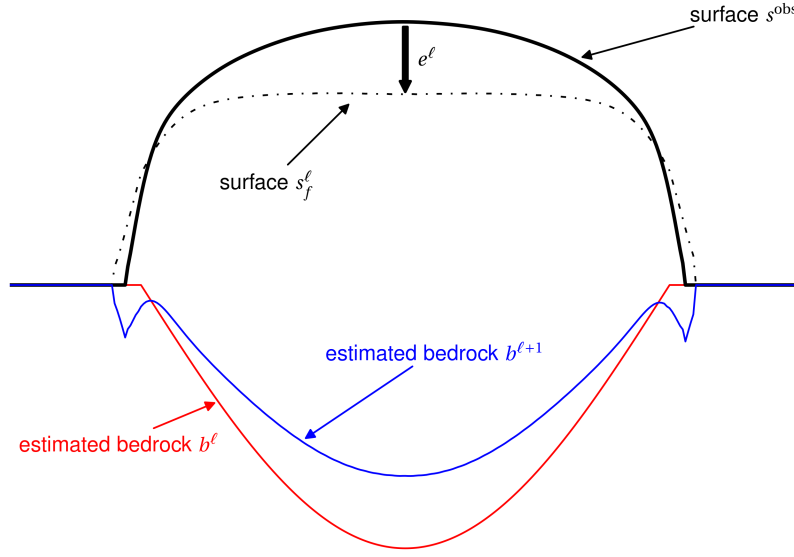


Figure 5.4 – Example glacier profile illustrating equation (5.33). The idea is to report the discrepancy $e^\ell = s^{\text{obs}} - s_f^\ell$ on the bedrock topography at each iteration. The bedrock is lifted where $e^\ell > 0$ and lowered where $e^\ell < 0$.

This section proposes a regularized fixed-point method that takes care of this drawback. Let $\ell > 0$ be an iteration index, b^ℓ the bedrock at fixed-point step ℓ and b^0 some initial guess for b . When no regularization applies, the idea is to iteratively update the bedrock with the surface topography misfit at final time, which is equivalent to superimpose the current bedrock b^ℓ and its updated version $s_f - \mathcal{H}^\ell|_{t=t_f}$ (see Figure 5.4):

$$b^{\ell+1} = b^\ell + \beta(s^{\text{obs}} - s_f^\ell) = b^\ell + \beta(s^{\text{obs}} - b^\ell - \mathcal{H}^\ell|_{t=t_f}) = (1 - \beta)b^\ell + \beta(s^{\text{obs}} - \mathcal{H}^\ell|_{t=t_f}), \quad (5.33)$$

where $s_f^\ell = b^\ell + \mathcal{H}^\ell|_{t=t_f}$ is the approximated final surface topography at iteration ℓ . In addition to this update, a smoothing of the computed ice thickness is necessary to allow for convergence. This is done by including a Laplacian term in equation (5.33). Whatever the forward model is, the procedure remains unchanged: First, the forward simulation is performed in order to determine the final surface topography. Then, the latter topography is compared to the measured topography s^{obs} and the misfit is added to the bedrock topography. The method is summarized in Algorithm 5.

Algorithm 5 Transient Inverse Method

initialize the bedrock topography with b^0

while $\|b^{\ell+1} - b^\ell\| > \delta$ **do**

solve the forward problem with bedrock topography b^ℓ and initial surface s_i , whose output is the ice thickness $\mathcal{H}^\ell|_{t=t_f}$ at final time;

solve

$$\begin{cases} \tilde{\mathcal{H}}^\ell = \mathcal{H}^\ell|_{t=t_f} + \beta(b^\ell + \mathcal{H}^\ell|_{t=t_f} - s^{\text{obs}}) + \varepsilon\beta\Delta\tilde{\mathcal{H}}^\ell, & \text{in } \Omega_\perp, \\ \tilde{\mathcal{H}}^\ell = 0, & \text{on } \partial\Omega_\perp, \end{cases} \quad (5.34)$$

for $\tilde{\mathcal{H}}^\ell$, where $0 < \beta \leq 1$ is a relaxation parameter and $\varepsilon \geq 0$ a regularization parameter;

set

$$b^{\ell+1} = s^{\text{obs}} - \tilde{\mathcal{H}}^\ell. \quad (5.35)$$

end while

In the two-dimensional, steady, shallow ice case, if $t_f - t_i = \Delta t$ and $\beta = 1$, Algorithm 5 is equivalent to solving problem (5.15). This approach is close to that exposed in [58], but here steady as well as transient problems are dealt with. The forward problem solved during the process can be any of the models that simulate the evolution of a glacier from initial time t_i to final time t_f .

This method aims at minimizing the surface topography misfit at time t_f without the use of its gradient. Note that, contrary to the procedure exposed in [26] and any shape optimization procedure, this method applies the smoothing explicitly, which does not make it an integral part of the algorithm. The algorithm is very easily implementable as it relies on the iterative use of the forward model. Moreover, it is as memory-demanding as the forward solver.

5.5 Numerical results

Various numerical results are presented in this section, most of which concern the two-dimensional algorithms, because the related computations are cheap. This section is exclusively devoted to the numerical validation of the methods on synthetical geometries inspired from real-world data. The inversion of real data is the topic of Section 5.7. All the numerical experiments presented below assume the availability of data satisfying the forward equation that is aimed at being inverted.

5.5.1 Stationary Inverse Method

For the sake of numerical validation, available data on two-dimensional glacier evolution according to model (2.32) are used. For instance, a set of bedrock and surface topographies with corresponding surface mass-balances that fulfill (2.32) with $\frac{\partial \mathcal{H}}{\partial t} = 0$ is assumed. The solution bedrock topography is denoted by b^* . Figure 5.5 shows the result of the inversion along principal flow lines for the Flask outlet glacier in Antarctica [33] and the Muragl, Silvretta, and Rhône mountain glaciers in the Swiss Alps. In particular, it illustrates how sensitive our test case geometries are to the regularization parameter. The results obtained using numerical scheme (5.19) are independent of the time step (see Table 5.1), as long as it satisfies a stability condition which seems to be of the form $\Delta t < c\Delta x^p$, for two real constants c and p . Furthermore, the method is independent of the mesh size (see Table 5.2): whatever the resolution of the original data is, the bedrock will be recovered with the same accuracy. Except on geometry (b), where the error is exceptionally larger when $\Delta x = 100$ m, the discrepancy with the solution bedrock b^* remains

essentially identical. For the sake of numerical comparison, the results presented in this section were calculated with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. Such a small value of δ is purely numerical and necessary for convergence. It is not linked with any glaciological quantity. In both tables, the large errors on geometry (a) are due to the large value of ε , as shown by Figure 5.5.

Table 5.1 – Maximal error $\|b - b^*\|$ in m between the computed and solution bedrock topographies for different time steps when $\varepsilon = 10^{-2}$ and $\Delta x = 25$ m with stopping $\delta = 10^{-6}$ m in the two-dimensional, stationary inverse method (5.17). When Δt is too large, the method does not converge, due to the stability condition of scheme (5.19). However, when the scheme is stable, the error remains identical for any time step. The large errors obtained for geometry (a) are due to the large value of ε (see Figure 5.5).

	Δt [a]			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Fig. 5.5(a)	$2.775e+02$	$2.776e+02$	$2.787e+02$	$2.907e+02$
Fig. 5.5(b)	$3.925e+00$	$3.925e+00$	$3.922e+00$	$3.901e+00$
Fig. 5.5(c)	$1.717e+00$	$1.741e+00$	$1.744e+00$	$1.773e+00$
Fig. 5.5(d)	$4.634e+01$	$4.637e+01$	$4.665e+01$	$4.984e+01$

Table 5.2 – Maximal error $\|b - b^*\|$ in m between the computed and solution bedrock topographies for different mesh sizes when $\varepsilon = 10^{-2}$ and $\Delta t = 10^{-3}$ a with stopping $\delta = 10^{-6}$ m in the two-dimensional, stationary inverse method (5.17). Except on geometry (b), where it is one order of magnitude larger on the mesh with lowest resolution than on the other meshes, the error remains essentially the same whatever the mesh size is.

	Δx [m]				
	100	50	25	12.5	6.25
Fig. 5.5(a)	$2.751e+02$	$2.752e+02$	$2.754e+02$	$2.766e+02$	$2.772e+02$
Fig. 5.5(b)	$2.014e+01$	$4.432e+00$	$3.925e+00$	$4.057e+00$	$3.917e+00$
Fig. 5.5(c)	$1.018e+00$	$1.386e+00$	$1.741e+00$	$1.697e+00$	$1.705e+00$
Fig. 5.5(d)	$4.248e+01$	$4.561e+01$	$4.637e+01$	$4.639e+01$	$4.639e+01$

Table 5.3 – Maximal error $\|b - b^*\|$ between the computed and exact thicknesses, number of iterations n_{its} , and computational time T_{CPU} required to obtain the results shown in Figure 5.5 for small enough regularization parameters in the stationary inverse method (5.17). Estimations were done with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. Note that these are only indicative results, as the number of iterations and the computational time depend strongly on the initial guess, the space and time steps, the regularization parameter and the stopping δ . The influence of ε on the number of iterations, and therefore on the computational time, is evinced by Figure 5.6.

	ε	$\ b - b^*\ $ [m]	n_{its}	T_{CPU} [s]
Fig. 5.5(a)	10^{-6}	$1.180e+00$	1527130	$3.580e+03$
Fig. 5.5(b)	10^{-4}	$5.973e-02$	96089700	$4.383e+04$
Fig. 5.5(c)	10^{-4}	$1.305e-01$	9613360	$3.093e+03$
Fig. 5.5(d)	10^{-3}	$2.101e+01$	13132500	$9.974e+03$

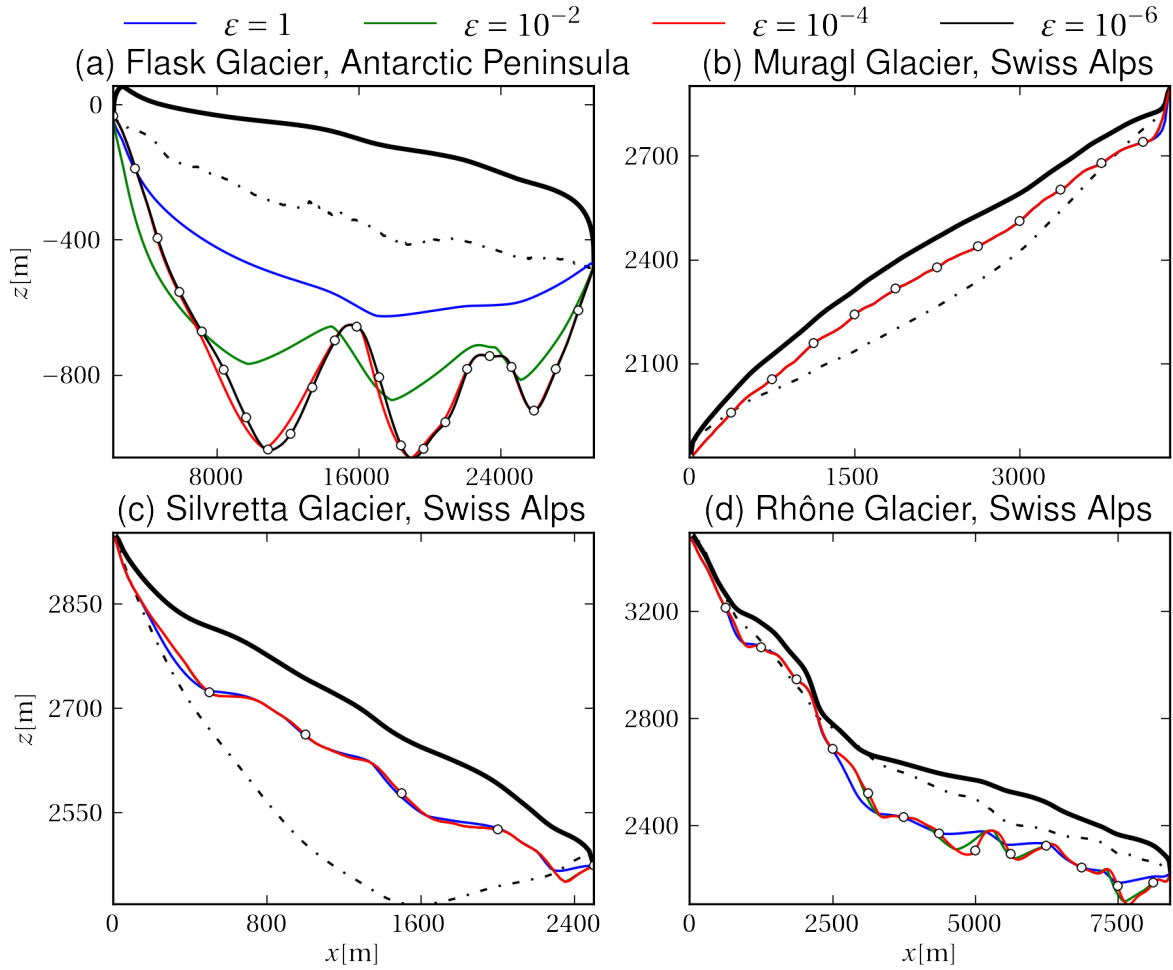


Figure 5.5 – Various examples illustrating the two-dimensional stationary inverse method (5.17), with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. In (a), we have a profile inspired of Flask Glacier that we modified in such a way that the thickness is zero at the boundaries. The stationary surface and the solution bedrock topographies are represented by the thick black line and the circles respectively. The dashed line stands for the initial guess of the subglacial topography and the colored lines delineate the resulting estimation for several regularization parameters. None of the geometries converged with $\varepsilon = 10^{-6}$ except geometry (a). Each time, for ε sufficiently small, the estimated topography is very close to the solution. The maximal error $\|b - b^*\|$ obtained between the solution and the estimation, the number of iterations until convergence, and the computational time are reported in Table 5.3 for reasonable regularization parameters. The influence of ε on the rate of convergence is illustrated by Figure 5.6.

Table 5.3 reports the maximal error $\|b - b^*\|$ between the computed and solution bedrock topographies, the number of iterations n_{its} needed to reach convergence, and the computational time T_{CPU} needed to achieve this convergence for small enough values of the regularization parameter ε . These last two quantities obviously depend on the initial guess, the space and time steps, the regularization parameter and the stopping criterion, so they are just indicative. Note that n_{its} corresponds to the number of computed time steps until condition (5.21) is met, where one such step consists in solving the non-linear equation (5.19) with a Newton method that usually converges after less than 5 iterations, which makes this implicit scheme fairly fast. The choice of δ depends on the glacier geometry and the chosen time step. Although we choose it very small compared to glaciological characteristic lengths, such a small value is necessary to get a reasonable solution, as illustrated by Figure 5.6. It shows how slow and indirect the convergence of the method is. For instance, on geometry (a), if one chooses the stopping $\delta = 10^{-3}$ m, then one cannot get a good estimation of the ice thickness. The way the method converges can be improved by increasing

the regularization ε .

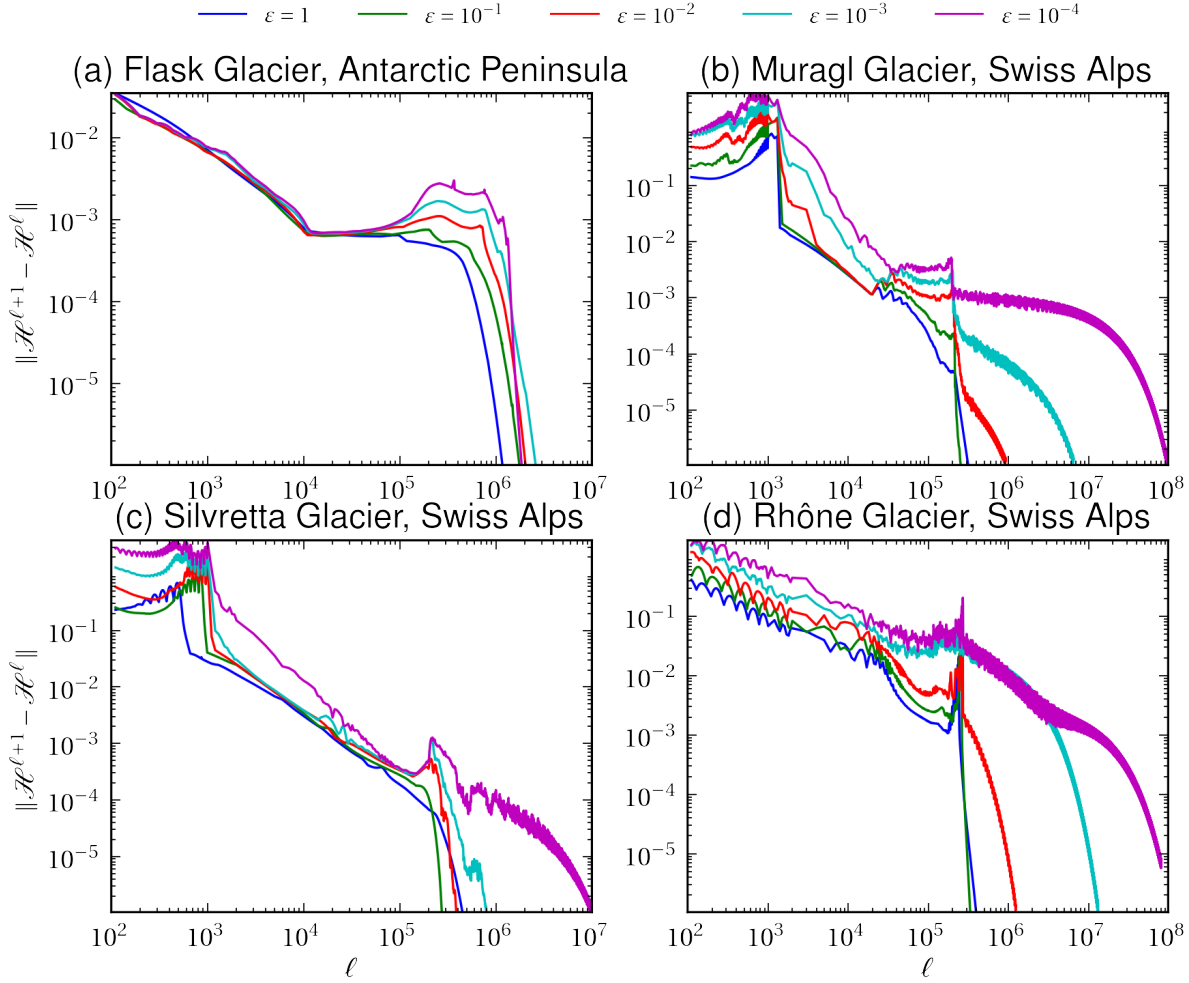


Figure 5.6 – Plots of $\|\mathcal{H}^{\ell+1} - \mathcal{H}^{\ell}\|$ as a function of the iteration number ℓ that show the sensitivity of the convergence rate of the stationary inverse method to the choice of the regularization ε in equation (5.17), with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. Geometry (a) shows how important it is to choose δ small enough in stopping criterion (5.21): a value close to $\delta = 10^{-3}$ m would give a bad estimation. Convergence is improved with increasing ε .

5.5.2 Quasi-stationary Inverse Method

Two-dimensional shallow ice flow

To validate the results with this approach, the algorithm is run on the geometries shown by Figure 5.5. The results are reported in Figure 5.7 when $t_f - t_i = 1$ a. The resulting errors when $t_f - t_i = 1, 5$ and 10 a are reported in Table 5.4. We notice that the larger the time frame $t_f - t_i$, the less accurate the estimation, as one would expect. Moreover, we see that the error depends on the surface discrepancy $\|s^{\text{obs}} - s_i\|$. The farther s^{obs} from s_i , the larger the error in the thickness estimation. Nevertheless, on these geometries that represent typical valley glaciers (except (a) that represents an outlet glacier in Antarctica), this first estimation is not far from the exact solution. The main interests of this method are its ease of implementation and reliability.

Table 5.4 – Maximal error $\|b - b^*\|$ in m between the computed and solution bedrock topographies and maximal surface topography discrepancy $\|s^{\text{obs}} - s_i\|$ in m as a function of the time frame $t_f - t_i$ in the two-dimensional quasi-stationary approximation (5.26) for the four different considered glaciers. Estimations were computed with $\Delta x = 25$ m and $\Delta t = 10^{-3}$ a with stopping $\delta = 10^{-6}$ m. Only the estimation over a time frame of 1 a is represented in Figure 5.7. We see that the error on the bedrock is very similar to the surface discrepancy in each case. The comparison holds as long as ε is small enough.

	$t_f - t_i = 1$ a		$t_f - t_i = 5$ a		$t_f - t_i = 10$ a	
	$\ b - b^*\ $	$\ s^{\text{obs}} - s_i\ $	$\ b - b^*\ $	$\ s^{\text{obs}} - s_i\ $	$\ b - b^*\ $	$\ s^{\text{obs}} - s_i\ $
Fig. 5.7(a), $\varepsilon = 10^{-6}$	$1.417e+00$	$1.002e+00$	$5.385e+00$	$4.985e+00$	$1.084e+01$	$9.897e+00$
Fig. 5.7(a), $\varepsilon = 10^{-3}$	$1.668e+01$	$3.711e+01$	$1.324e+01$	$3.612e+01$	$1.600e+01$	$4.310e+01$
Fig. 5.7(a), $\varepsilon = 10^{-4}$	$8.234e-01$	$9.733e-01$	$4.145e+00$	$5.193e+00$	$7.494e+00$	$1.142e+01$
Fig. 5.7(a), $\varepsilon = 10^{-4}$	$3.981e+00$	$4.957e+00$	$1.297e+01$	$1.938e+01$	$1.894e+01$	$3.086e+01$

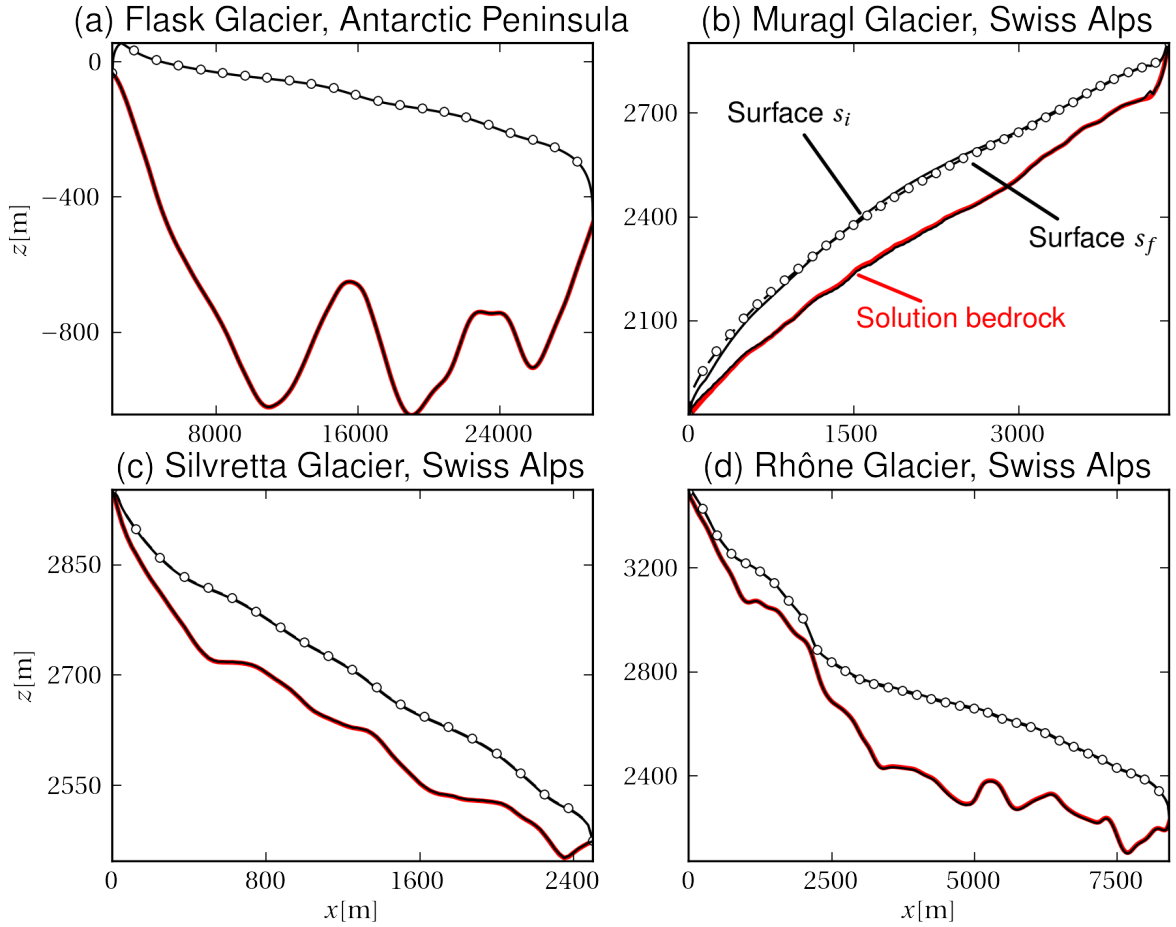


Figure 5.7 – Various examples illustrating the two-dimensional quasi-stationary inverse method (5.26), with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-3}$ m. The black line stands for the initial surface geometry, the dots represent the target surface after 1 a, the thick red line delineates the solution bedrock and the thin black line near the solution is the quasi-stationary estimation with data over a time frame of 1 a. Even the estimation of irregular bedrocks is good. The maximal errors between the solution bedrock and its estimation as a function of $t_f - t_i$ is reported in Table 5.4, with the value of the applied regularization ε . The largest error is obtained on geometry (b), where the difference between s_i and s^{obs} is the largest.

Three-dimensional shallow ice flow

To validate Algorithm 4, a set of bedrock and surface topographies as well as a surface mass-balance time series \mathcal{B} are assumed that satisfy problem (2.31) on the two geometries of Silvretta and Gries glaciers with sliding included over a time frame of 1 a. On both glaciers, the same sliding parameters are set to $z_{sl} = 2700$ m and $\mathcal{C} = 0.025 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ (colored, contour lines on the large contour plot of Figures 5.11 and 5.10). Note that these values do not correspond to the values tuned with field data, the purpose here being only to validate the method on synthetical data. Only the glaciers' geometries are inspired from real-world measurements. In this validation, one assumes that the surface topography is available at times $t_i = 0$ and $t_f = 1$ a. Moreover, the time and space steps are set to $\Delta t = 10^{-4}$ a and $dx = 50$ a respectively and the regularization $\varepsilon = 1$. The ice thickness is initialized by solving the Laplace equation (4.2) inside the glaciers' outline with such a right-hand side that the maximal thickness is 300 m on Gries and 200 m on Silvretta glacier. After a few millions of iterations, *i.e.* a few minutes of computational time, an estimation is obtained in both cases. In the following sections, the surface and bedrock topography misfits are compared and analyzed. A “topography misfit” is the half of the sum of the square of the difference between the computed and exact, or measured, topography over the whole computational domain (see expression (6.33)). Figure 5.8 illustrates how the surface and bedrock topography misfits evolve with the iteration number. First, the surface topography misfit is decreased, before it reaches a minimum value, that is still high. The bedrock misfit functional first decreases and then increases until it attains some value that is also high. During the whole increase in bedrock topography misfit, the surface misfit remains unchanged. Figure 5.9 shows with black markers the comparison of the exact to the initial ice thicknesses on Gries (left) and Silvretta (right) glaciers. The red markers show the comparison of the exact to the computed ice thicknesses. Ideally, all the markers should follow the dashed line representing all the points such that the computed thickness with the quasi-stationary inverse method is exactly equal to the exact thickness. Here, this is only approximatively true. The computed estimation is however good, although the method is very basic. The estimation is better on Silvretta glacier, which is thinner. For the sake of visualization, Figures 5.11 and 5.10 display selected profiles of the estimated bedrock topography across the glacier (dashed curves) along with the solution bedrock (thick, red curves) and target surface topographies (thick, black curves). With the size of the time frame, the error made in the bedrock estimation is of the same kind as with the two-dimensional method presented above. On both Figures, a small surface topography discrepancy is noticeable compared to the bedrock topography's.

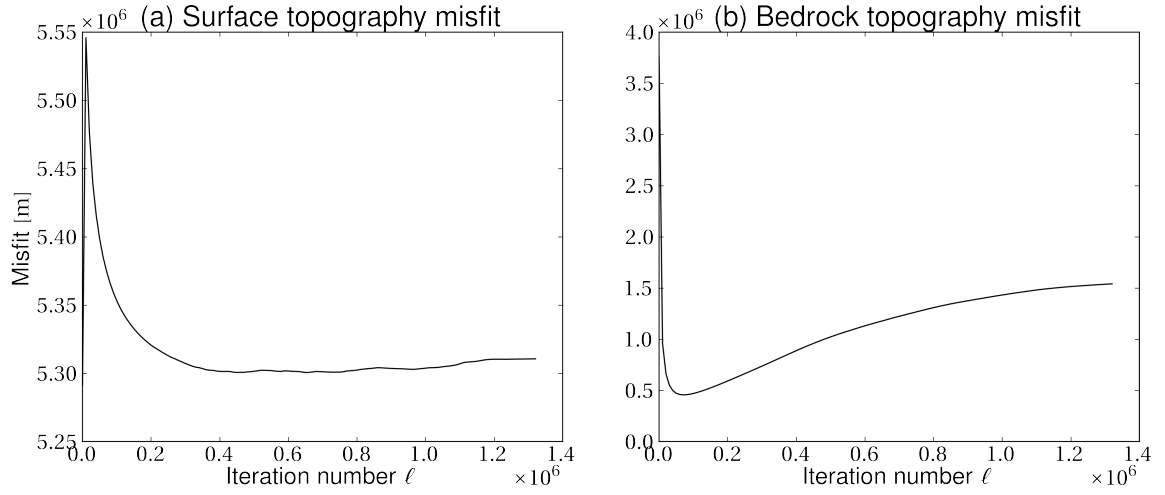


Figure 5.8 – Surface and bedrock topography misfits as functions of the iteration number on Silvette glacier when the three-dimensional, quasi-stationary inverse method is applied, over a time frame of 1 a. The simulation was performed with space and time steps $\Delta x = 50$ m and $\Delta t = 10^{-4}$ a respectively and a regularization $\varepsilon = 1$. The misfit represented here is the square of the L_2 -norm of the topographies' discrepancy (see expression (6.33)). The surface topography misfit reaches a minimal value, that is still high. During the process, the bedrock topography misfit first decreases, before it increases again. The maximum deviation in the bedrock topography is 58 m, while it is only 17 m in the final surface topography.

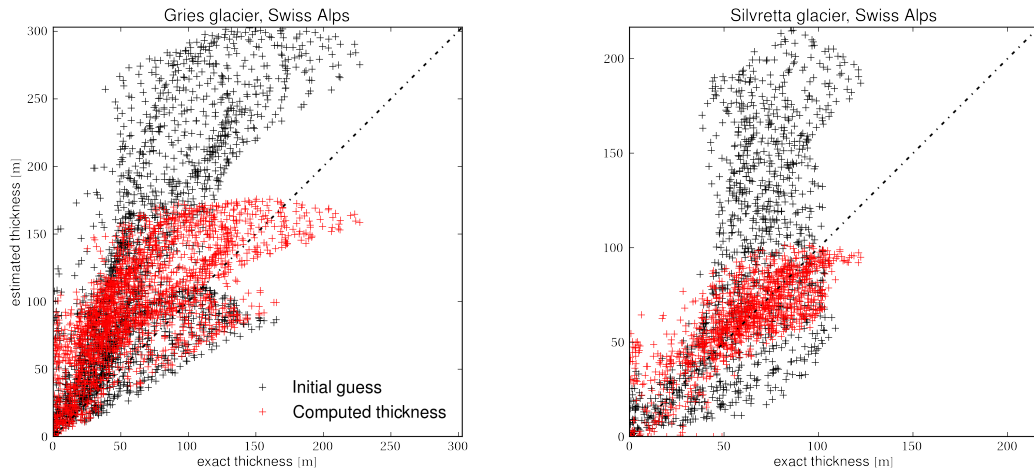


Figure 5.9 – Ice thickness comparisons on Gries (left) and Silvette (right) glaciers. A time evolution over 1 a is considered. The space and time steps are set to $\Delta x = 50$ m and $\Delta t = 10^{-4}$ a respectively and the regularization to $\varepsilon = 1$. The black markers depict the comparison of the thickness' initial guess to the exact ice thickness, while the red markers show the the comparison of the computed ice thickness with the quasi-stationary inverse method (5.28) to the exact thickness. Results are slightly better on Silvette glacier. Note the difference of scales. If the method was perfect, then all the red markers should match exactly with the dashed line, which represents the points such that the exact thickness equals the computed thickness.

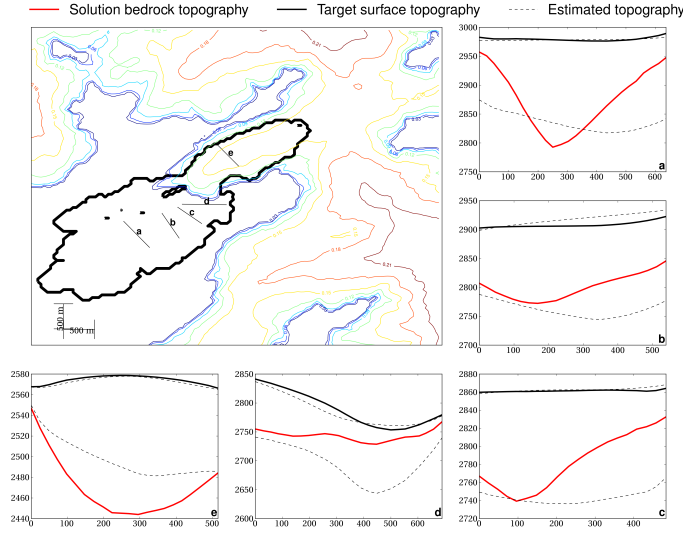


Figure 5.10 – Illustration of the bedrock topography estimated with the quasi-stationary inverse method (5.28) on Gries glacier that complements the global Figure 5.9(a). The regularization parameter is $\varepsilon = 1$. The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot.

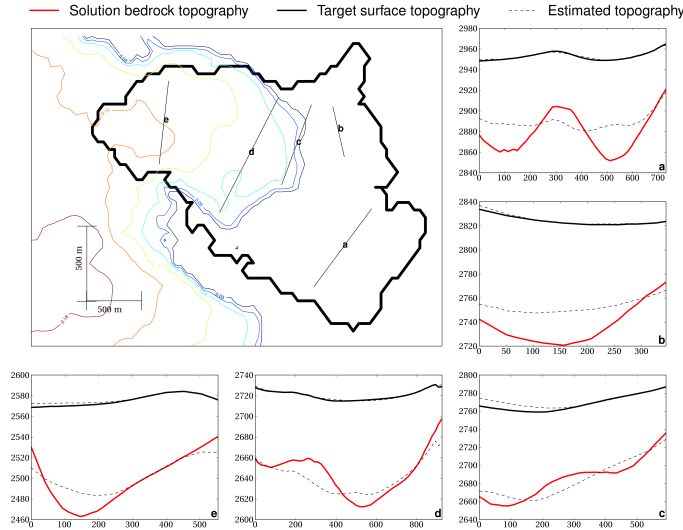


Figure 5.11 – Illustration of the bedrock topography estimated with the quasi-stationary inverse method (5.28) on Silvretta glacier that complements the global Figure 5.9(b). The regularization parameter is $\varepsilon = 1$. The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot. The dependence of the result in the regularization is analogous to the two-dimensional case (see Figure 5.5).

5.5.3 Transient Inverse Method

Two-dimensional shallow ice flow

None of the numerical experiments converged without regularization, except on some steady geometries, on very small time frames. Obviously, although the algorithm converges, it is not possible to exactly get $s^{\text{obs}} = \lim_{\ell \rightarrow \infty} s_f^\ell$ since the problem is slightly modified by the regularization. As in the stationary case, the true solution to the transient problem cannot be found in general, but can be approximated.

Tests on time intervals of 1 a, 5 a, and 10 a show that the errors remain stable with the size of the time frame, as reported in Table 5.5. Moreover, they are much smaller than those of Table 5.4 in the quasi-stationary approximation, except on geometry (c) where a Gaussian filter had to be applied to b^ℓ after each iteration to make the algorithm converge.

Finally, the computational time and the number of iterations needed to converge when $\varepsilon = 100$ are reported in Table 5.6. Compared to the reference values for stationary and quasi-stationary estimations listed in Table 5.3, they are small, except on geometry (a), for smaller errors (compare to Table 5.4). Hence this two-dimensional fixed-point procedure is more accurate than the previous ones on our test case mountain glaciers.

Table 5.5 – Maximal error on the geometries of Figure 5.7 for various time frames and regularization ε with the two-dimensional transient inverse method. Estimations were computed with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, $\beta = 0.1$, and $\delta = 10^{-6}$ m. Geometry (c) did only converge with Gaussian filtering of the estimated bedrock topography after each iteration. Note that these errors are much smaller than those of Table 5.4 in the quasi-stationary approximation.

	$t_f - t_i = 1$ a			$t_f - t_i = 5$ a			$t_f - t_i = 10$ a		
	$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$	$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$	$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$
Fig. 5.7(a)	8.6e-01	5.1e-01	2.7e+00	8.7e-01	4.8e-01	2.8e+00	9.2e-01	4.6e-01	2.8e+00
Fig. 5.7(b)	5.3e-02	2.9e-01	2.3e+00	5.2e-02	3.0e-01	2.3e+00	5.2e-02	2.8e-01	2.2e+00
Fig. 5.7(c)	6.9e+01	6.9e+01	7.0e+01	1.2e+01	1.2e+01	1.4e+01	5.5e+00	5.5e+00	5.7e+00
Fig. 5.7(d)	2.9e-01	5.7e-01	4.1e+00	3.0e-01	5.7e-01	2.9e+00	2.9e-01	5.7e-01	2.8e+00

Table 5.6 – Number of iterations and computational time needed by the transient inverse method to converge with $\varepsilon = 100$, $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m.

	$t_f - t_i = 1$ a		$t_f - t_i = 5$ a		$t_f - t_i = 10$ a	
	n_{its}	T_{CPU} [s]	n_{its}	T_{CPU} [s]	n_{its}	T_{CPU} [s]
Fig. 5.7(a)	17493	16510	17309	69957	17286	148299
Fig. 5.7(b)	925	218	853	707	797	1241
Fig. 5.7(c)	950	204	611	377	374	396
Fig. 5.7(d)	2500	926	2459	3458	2444	6540

Three-dimensional shallow ice flow

The behavior of the method is the same as its two-dimensional version, save that it now supplies a value for the glacier's ice thickness everywhere in the map domain. For the sake of comparison with the quasi-stationary inverse method presented in the previous section and the shape optimization algorithm formulated in the next chapter, convergence plots of the surface and bedrock topography misfits are depicted by Figure 5.12. In the shallow ice approximation, the transient inverse method converges on the tested geometries. Contrary to the bedrock topography misfit in the quasi-stationary inverse method, the misfit in the transient inverse method is decreased

during the whole inversion process. The comparison between the computed and exact ice thicknesses is depicted by Figure 5.13. Compared to its quasi-stationary counterpart, the transient inverse method provides an estimation where the markers are located around the dashed- line more uniformly.

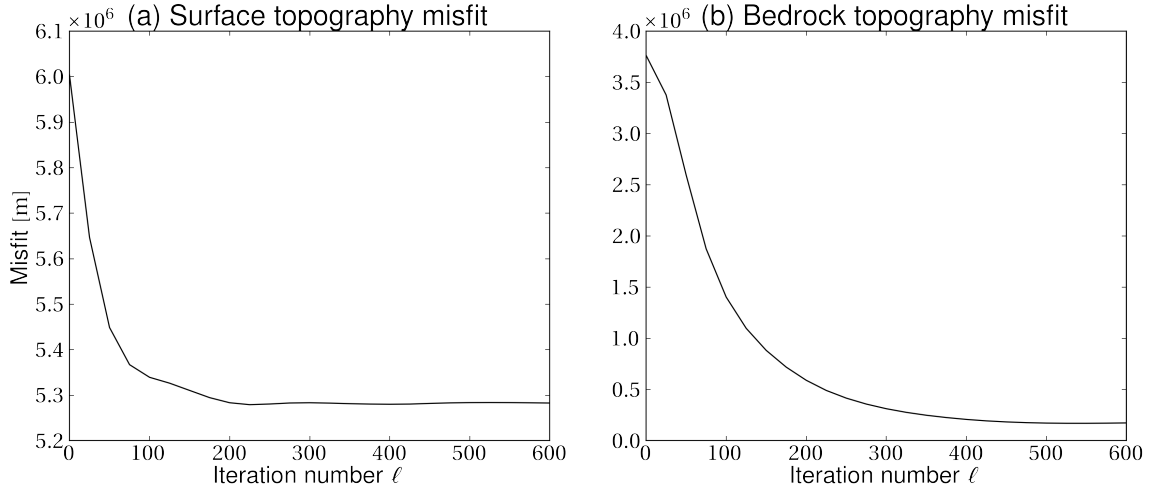


Figure 5.12 – Surface and bedrock topography misfits as functions of the iteration number in the three dimensional, shallow ice, transient inverse method. The misfit is the square of the L_2 -norm of the topography misfit (see equation (6.33)). A time evolution over 1 a is considered with a regularization parameter $\varepsilon = 100$. Contrary to its behavior in the quasi-stationary inverse method (see Figure 5.8), the bedrock topography misfit decreases before it attains a minimal value and does not increase after that. The value of both misfits is still high with this method. The maximal deviation of the computed bedrock from the solution is 75 m, while is only 15 m for the surface topography.

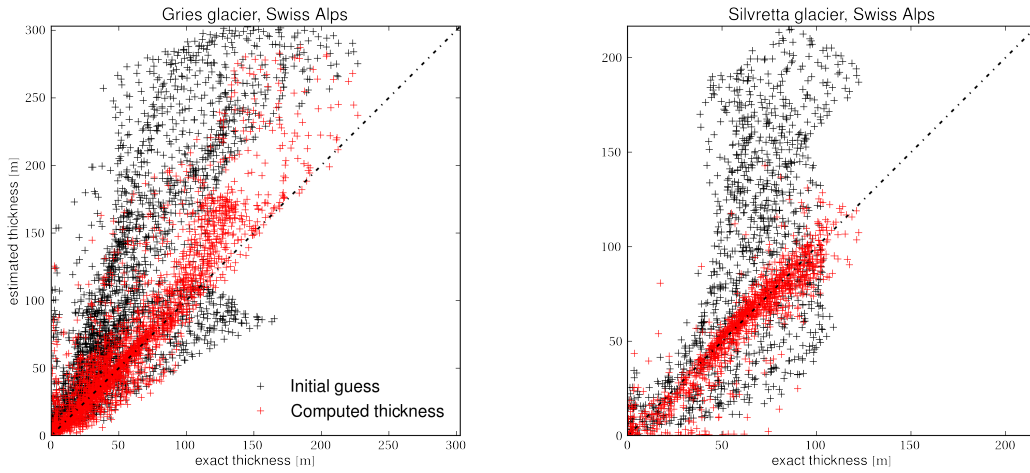


Figure 5.13 – Ice thickness comparisons on Gries (left) and Silvretta (right) glaciers. A time evolution over 1 a is considered. The regularization parameter is $\varepsilon = 100$. The black markers depict the comparison of the thickness' initial guess to the exact ice thickness, while the red markers show the the comparison of the computed ice thickness with the transient inverse method (Algorithm 5) to the exact thickness. Compared to the results of the quasi-stationary inverse method (see Figure 5.9), these results are better in the sense that the red markers are distributed more uniformly around the dashed line that represents all the points such that the exact thickness is equal to the computed thickness. The dependence of the result in the regularization is analogous to the two-dimensional case (see Table 5.3).

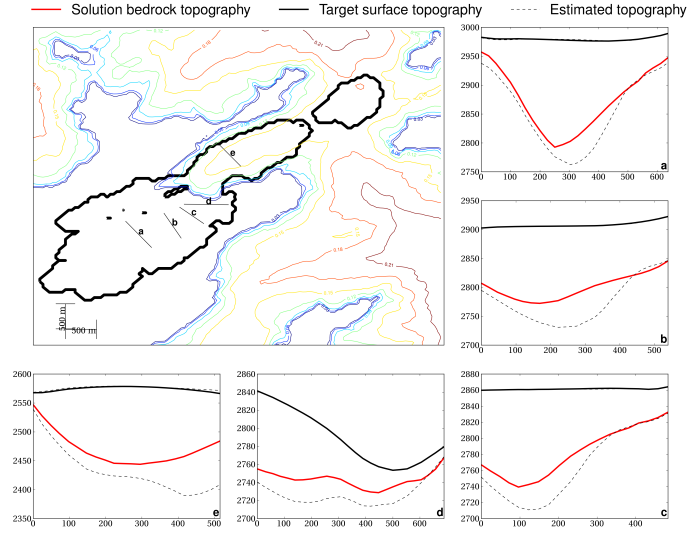


Figure 5.14 – Illustration of the bedrock topography estimated with the transient inverse method (Algorithm 5) on Gries glacier that complements the global Figure 5.13(a). The regularization parameter is $\varepsilon = 100$. The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot.

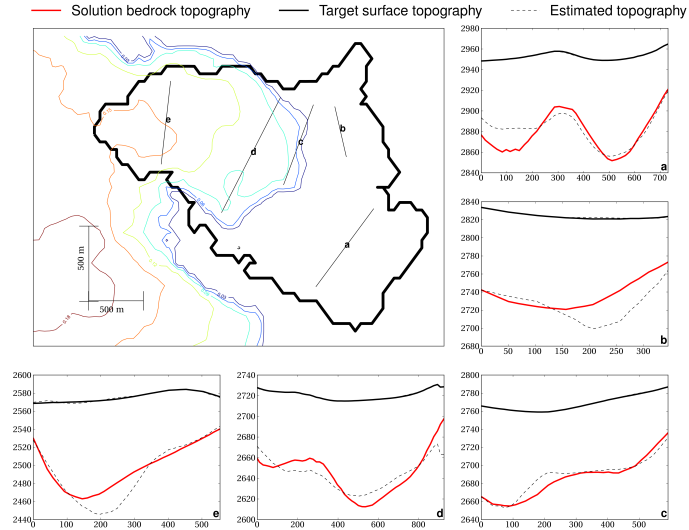


Figure 5.15 – Illustration of the bedrock topography estimated with the transient inverse method (Algorithm 5) on Silvretta glacier that complements the global Figure 5.13(b). The regularization parameter is $\varepsilon = 100$. The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot. The dependence of the result in the regularization is analogous to the two-dimensional case (see Figure 5.5).

Three-dimensional Stokes ice flow

Although it is the straightforward extension of the transient inverse methods formulated previously and actually aims at the minimization of the surface topography misfit functional, this method is slow and not convergent. This is because the Stokes ice responds differently to bedrock topography perturbations than its shallow ice counterpart, as explained in Chapter 4. The analysis below refers always to the following numerical experiment: starting from the bedrock and initial surface topographies of Silvretta glacier, first simulate its evolution over 10 a. Such a long time frame is chosen because the inversion didn't perform well on a time frame of 1 a as in the previous sections in the shallow ice approximation. Next, compute the solution of the Laplace equation

$$\Delta \mathcal{H} = c \quad (5.36)$$

on the domain where the ice thickness \mathcal{H} is supposed unknown, where c is such a constant that the maximal thickness is 150 m. Then, apply the TIM to this perturbed geometry and compare the obtained bedrock topography to the original one. Figure 5.16 shows the error curves resulting from a typical simulation on Silvretta glacier with a structured, anisotropic, finite-element mesh and a structured mesh of cubic cells of sizes $(H, h) = (50, 10)$ m. The time frame is 10 a. Figure 5.16(a) shows that after some iterations, the bedrock topography is not improved any more, as the surface topography misfit. Figure 5.16(b) illustrates the convergence of the method. Instead of converging, the difference between two successive bedrock topographies oscillates around some value.

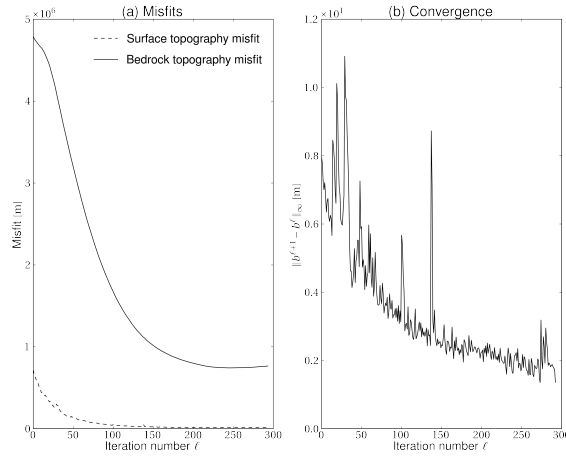


Figure 5.16 – Convergence plots resulting from the TIM calculation on Silvretta glacier, starting from such a bedrock that the initial thickness is 300 m, over a time frame of 10 a. A reasonable solution is determined after about 250 iterations. (a) Surface (dashed line) and bedrock (continuous line) topography misfits, *i.e.* square of the L_2 -norm of the discrepancy between the exact and computed topographies (see expression (6.33)) during the first 300 iterations. (b) Bedrock topography maximal changes from one iteration to the other as a function of the iteration number. After this inversion, the maximal bedrock topography deviation with the solution is 92 m which the maximal final surface topography deviation is only 14 m.

Figure 5.17 depicts the obtained estimation along several profile lines. The large, top left plot shows the contour of Silvretta glacier. The smaller subplots compare the bedrock and final surface topography estimations to the corresponding solutions. A good agreement between the solution and estimated bedrock topographies can be noticed. Up-glacier, namely on profiles **a**, **b**, and **c**, the surface topography misfit is very small. The estimation is worse down-glacier, near the tongue (profile **e**).

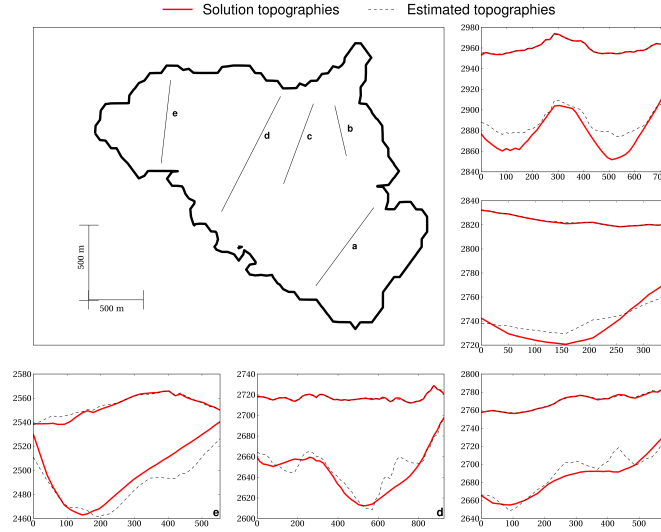


Figure 5.17 – Bedrock and final surface topography estimations along profile lines on Silvretta glacier, Swiss Alps. Inside the glacier’s contour (top left) are shown the selected profile lines. Ice flows from profile **a** to profile **e**. The thick, continuous, red lines represent the solution topographies, while the dashed, black ones the estimated topographies. On each subplot, the top curves delineate the surface topographies and the bottom curves the bedrock topographies. Up-glacier (profiles **a**, **b**, and **c**), the surface topography misfit is very small, although the bedrock topography misfit is about 20 m. Down-glacier (profile **e**), the bedrock topography misfit is larger, as is the surface topography’s.

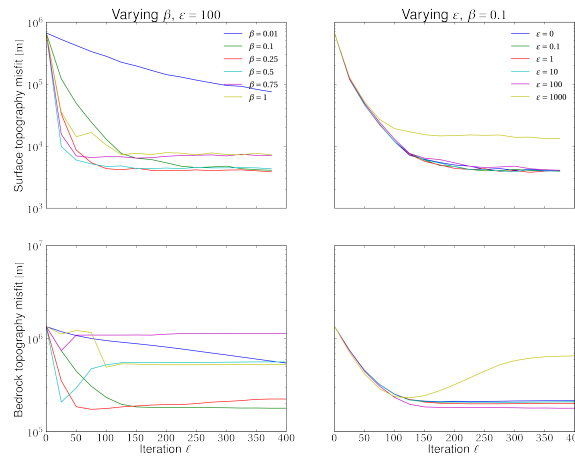


Figure 5.18 – Surface (first row) and bedrock (second row) topography misfits as functions of the iteration number ℓ for different values of the relaxation (left column) and regularization (right column) parameters β and ϵ respectively. Recall that the misfit is the square of the L_2 -norm of the topographies’ discrepancy (see expression (6.33)). The curve with $\beta = 0.01$ is represented only to show how many more iterations the algorithm needs with such a small parameter. The plots corresponding to varying β correspond to estimations with $\epsilon = 100$ while the other ones to estimations with $\beta = 0.1$. Too large values of β are responsible for the jumps in the convergence curves. In turn, very large values of ϵ make the topography misfits increase. The regularization parameter does not affect much the behaviour of these curves, except if too large. Again, it is noticeable that the surface topography misfit is much smaller than the bedrock topography’s (about one order of magnitude in these plots).

The Algorithm 5 makes use of a regularization and relaxation parameter ϵ and β respectively. Figure 5.18 shows the effects of the two parameters on the evolution of the surface and bedrock topography misfits with the iteration number. The largest differences occur with varying β . A large relaxation parameter makes the algorithm attain the minimum of the surface topography misfit after fewer iterations but leads to more instabilities. To the contrary, the regularization ϵ does not affect much the behaviour of these curves, except for extremely large values. However, as shown by Figure 5.19, it smooths the bedrock topography.

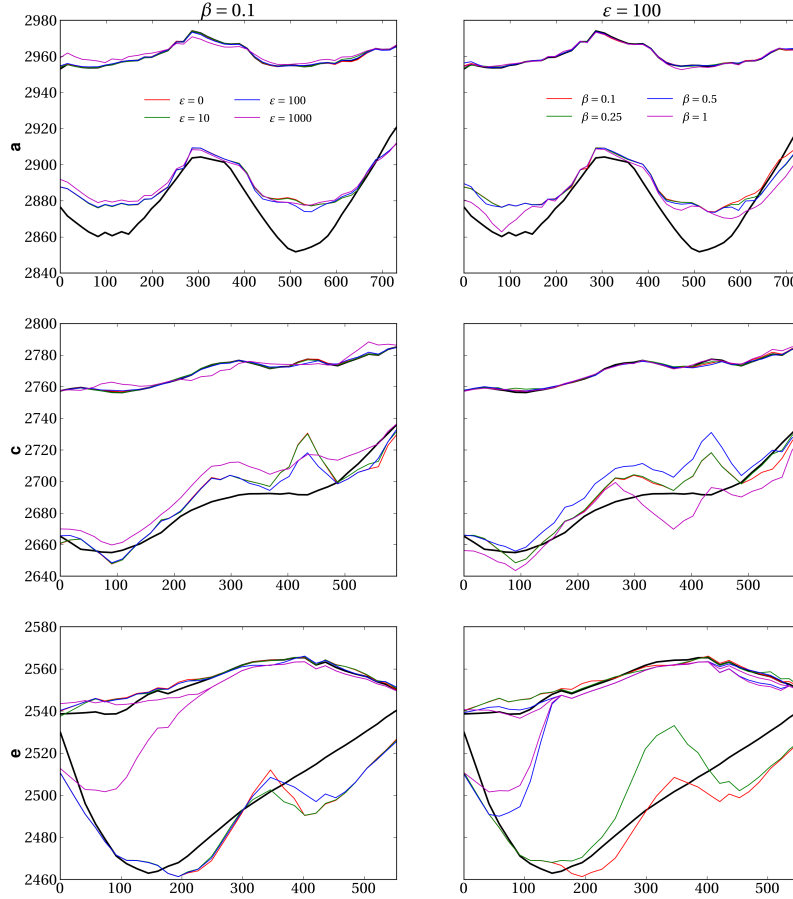


Figure 5.19 – Bedrock and surface topography estimations along the profile lines **a**, **c**, and **e** depicted on Figure 5.17. The first column shows the estimations obtained with $\beta = 0.1$ and various regularization parameters ϵ . The second column of plots shows the estimations obtained with $\epsilon = 100$ and various relaxation parameters β . The different estimations are plotted with different colors. The solid black lines represent the solution topographies, the top curve standing for the target surface topography and the bottom curve for the solution bedrock topography. Large values of ϵ make the bedrock topography smoother but also farther from the solution. With large values of β , the algorithm attains the minimal surface topography misfit sooner but is also more unstable.

Model intercomparison

Because the evaluation of estimation (5.10) or other low order, two-dimensional approximations of ice flow is very fast, it is usually applied in the literature to a set of flow lines previously determined. Eventually, the results are interpolated to obtain a Digital Elevation Model (DEM) of the subglacial topography. There are many ways to achieve

this interpolation, as for example with the Radial Base Function approach [19], the Inverse Distance Weighting technique [108], the Natural Neighbour interpolation [77], or even the assumption that glaciers have parabolic cross-sections [75]. A more physical method was proposed in [63, 64, 76]. Although fast, all these interpolation methods introduce non-physical data into the final estimation, which makes them inaccurate.

Let us therefore compare the subglacial topography of Muragl glacier obtained with the transient inverse method based on the two-dimensional shallow ice approximation (equation (2.32)), the three-dimensional shallow ice approximation (equation (2.31)), and the Stokes approximation (equations (2.27)-(2.8)) on synthetic data. Let us assume three-dimensional topographic and mass-balance data are available on Muragl glacier that evolve according to the Stokes approximation (2.27)-(2.8). This numerical experiment is purely artificial and does not involve measured surfaces topographies. Then, let us suppose the bedrock topography is unknown and try to estimate it first from the Natural Neighbour interpolation [77] of two-dimensional profile data along the flow lines shown on Figure 5.20. Second, let us calculate it by inverting the three-dimensional SIA and, finally, by inverting the three-dimensional Stokes approximation. The shallow ice estimations were performed with a space step $\Delta x = 25$ m and the estimation with the Stokes approximation was done with finite elements of horizontal size 25 m and vertical size 6.25 m. The results along the selected profile lines are shown on Figure 5.20. The estimation is satisfactory in each case, except flow line *e* in the three-dimensional shallow ice approximation that seems to underestimate the thickness at the glacier's tongue. In the literature, this is sometimes corrected by introducing sliding in the shallow ice approximation and tuning the sliding parameters such that the computed surface topography at final time fits the measured topography. This numerical experiment demonstrates that such a fitting is clearly unphysical, as Muragl glacier was simulated without sliding. Along transversal lines, shown on Figure 5.21, it can be noticed that the accuracy of the lower order models is far from the one of the Stokes approximation. This is confirmed by Figure 5.22 that compares the exact ice thickness to the computed one. The shallow ice estimations have a maximal deviation of 50 m from the solution, while the maximal deviation of the Stokes estimation is 25 m.

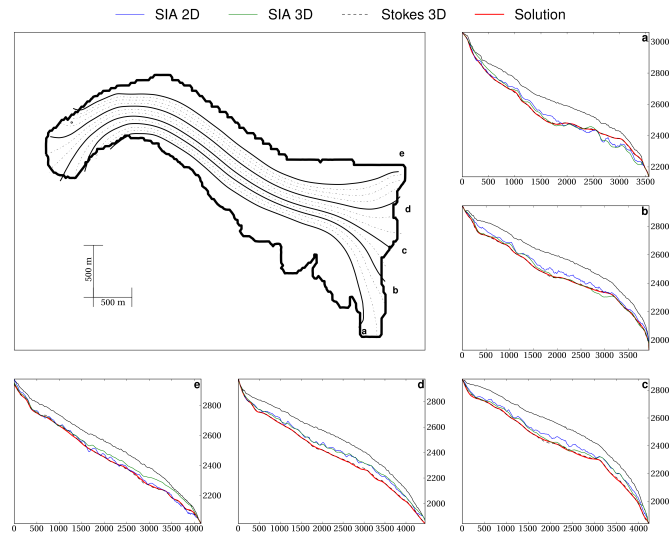


Figure 5.20 – Results along selected flow lines of Muragl Glacier. The two-dimensional computations were done along the continuous and dashed flow lines. The profiles we show correspond to those along the thick flow lines only. The estimations are satisfactory in each case, except on flow line *e*, where the SIA 3D underestimates the thickness at the tongue.

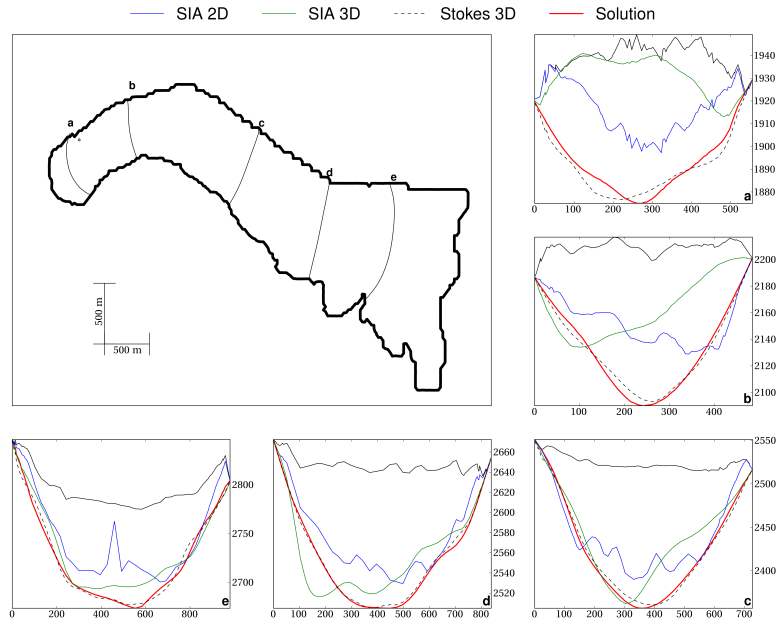


Figure 5.21 – Results along profiles transverse to the central flow line of Muragl Glacier. Note the change of scale compared to Figure 5.20. We see a higher accuracy of the higher order Stokes estimation, even near the tongue.

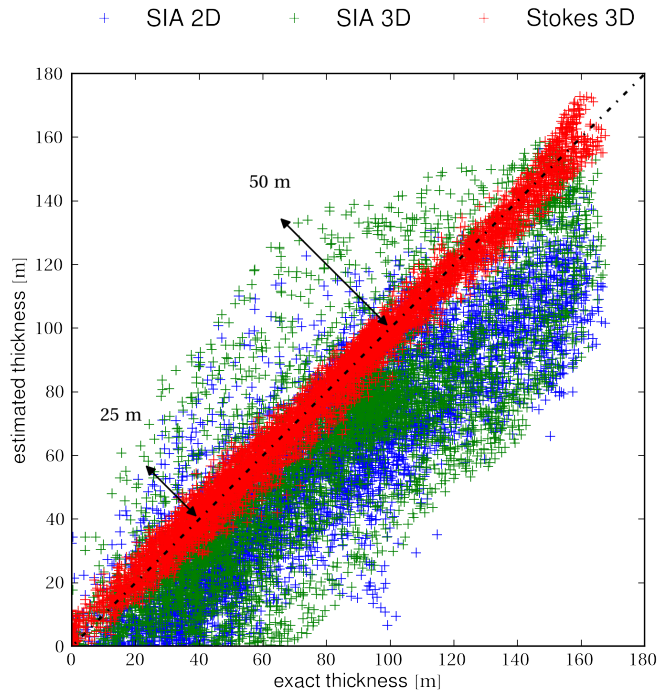


Figure 5.22 – Comparison of exact (x -axis) and computed (y -axis) ice thickness for Muragl Glacier. Without surprise, the best results are given by the Stokes approximation of ice flow. The thickness distributions in the SIA make a cloud of size 50 m around the identity function (dash-dotted line).

5.6 Sensitivity analysis

The input data used in the inversion approaches of this chapter are the glacier surface topographies and the surface mass-balance time series. The methods based on the shallow ice approximation are much more sensitive to surface topography than to surface mass-balance [85]. This is essentially because the thickness depends on the derivative of the surface topography and the integral of the surface mass-balance. In this section, instead of a systematic sensitivity analysis, an analysis restricted to surface topography and mass-balance perturbations of the form $a\chi(x)$ is first presented in the two-dimensional, shallow ice, quasi-stationary and transient inverse methods, where the perturbation's amplitude $a > 0$ is of the order of magnitude of typical measurement errors and $\chi(x)$ is a uniformly distributed random variable on the interval $[-1, 1]$. The sensitivity of the three-dimensional shallow ice methods is identical. Then, a short sensitivity analysis of the transient inverse method for Stokes ice flows is exposed.

5.6.1 Two-dimensional shallow ice flows

The stationary and the quasi-stationary methods are both similarly sensitive to surface perturbations. Figure 5.23 shows the resulting estimations in the quasi-stationary approximation (5.26) when the surface topographies undergo perturbations with $a = 1$ m. The graphics resolution cannot highlight the difference between the original and the perturbed surfaces. However, it can clearly be noticed on the resulting estimation, even with the regularization.

Smoothing the input surface does not improve the results in the stationary and quasi-stationary models. Numerical experiments run with Gaussian filtered data yield only slightly less oscillating estimations. In two dimensions, the data oscillations could be controlled with the truncation of their highest Fourier modes [58], but this is not possible any more in three dimensions. The best option consists in smoothing the converged estimation.

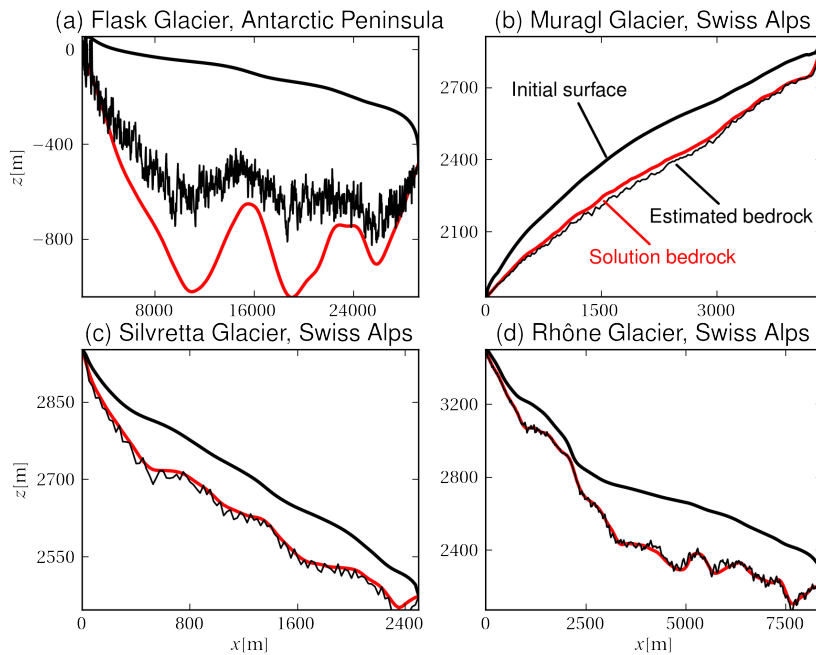


Figure 5.23 – Sensitivity of the method to surface topography in the quasi-stationary case, for a time frame of one year and perturbations with $a = 1$ m. The thick black and red lines show the original, unperturbed surface and solution respectively, while the thin black line stands for the converged estimated bedrock with the perturbed data. Regularization parameter used: (a) $\varepsilon = 10^{-6}$ (b) $\varepsilon = 10^{-2}$ (c) $\varepsilon = 10^{-4}$ (d) $\varepsilon = 10^{-4}$.

With the transient inverse models, no convergence can be achieved without at least a Gaussian filtering of the input surfaces and the bedrock topographies. Then, the regularized method with ε of the order 10^3 coupled with the Gaussian filtering of the bedrock estimation at each iteration is necessary to obtain convergence close to the solution (see Figure 5.24). The method is then dependent on the filtering. As in the quasi-stationary approximation, the results are satisfactory for any time frame, except in the outlet-glacier case (Figure 5.24(a)). This may be due to the relatively large thickness, which influences the inversion by dispersing the surface information. At least, the bumpy shape of its bedrock topography can be predicted. Globally, this last method yields better results than the quasi-stationary approximation.

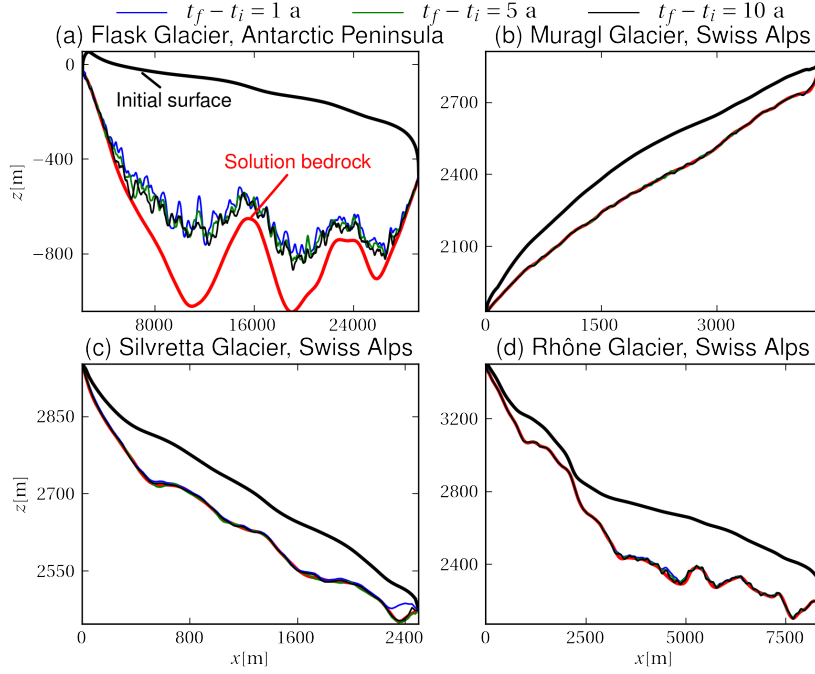


Figure 5.24 – Sensitivity of the transient inverse method to surface topography for various time frames and perturbations with $a = 1$ m. The thick black and red lines show the original, unperturbed initial surface and solution, while the thin blue line stands for the estimated bedrock over 1 a, the green line for the estimation over 5 a, and the black line for the estimation over 10 a. A Gaussian filter was applied after each iteration. Errors are significantly lower than in the quasi-stationary approximation. Regularization parameter used: (a) $\varepsilon = 10^3$ (b) $\varepsilon = 1$ (c) $\varepsilon = 10^3$ (d) $\varepsilon = 10^3$.

On the other hand, numerical experiments were run to test for the sensitivity to the surface mass-balance, with a ranging from 0 to $2\mathcal{B}$. Although these perturbations are extremely large and oscillatory, all the methods converge well. No additional Gaussian filtering is even needed in the transient inverse method for transient problems. As long as the applied surface mass-balance oscillates around the original one, the estimation does not suffer from the perturbation.

5.6.2 Three-dimensional Stokes ice flows

Forward, shallow ice computations always end up with smooth surface topographies. Hence, it is not surprising that the input to the inverse shallow ice methods must be smooth. However, the surface topographies arising from the simulation of Stokes ice flows are not smooth (see *e.g.* Figure 3.6). Therefore, the transient inverse method for Stokes flows is not very sensitive to noise in the data. However, it is interesting to investigate its susceptibility to the other model parameters. An illustration is given by Figure 5.25, where the sensitivity of the transient inverse method for Stokes flow is reported for the rate factor, the surface topography, the melting rate, and the sliding constant on

the geometry of Silvretta glacier. The numerical experiments that lead to this Figure are the following: first, simulate the glacier's evolution over 10 a with the forward model (2.8)-(2.27). Then, apply the transient inverse method (Algorithm 5) with either the rate factor, or the surface topography, or the melting rate, or the sliding constant wrong, to see their influence during the inversion process. The observed surface topography is replaced with the linear combination

$$s = \epsilon s_i + (1 - \epsilon) s^{\text{obs}}, \quad 0 \leq \epsilon \leq 1. \quad (5.37)$$

On Figure 5.25, b_{ref} is the bedrock topography obtained via the transient inverse method applied with the true parameters. Each subplots compares the computed bedrock topography with a wrong parameter to the reference bedrock b_{ref} . On this Figure, the sensitivity to the sliding and equilibrium line altitudes z_{sl} and z_{ELA} is not displayed, because the method does not supply relevant results with different values of these parameters. Most of the time, the surface topography misfit doesn't decrease at all.

From Figure 5.25, one notices that the method's sensitivity to the rate factor, melting rate, and sliding constant is much higher than the forward problem's (compare to Figures 4.2(a), (c), and (f)). Moreover, the discrepancies are distributed in another way. For example, the maximal discrepancy with a too low melting rate is reached up-glacier whereas it was attained down-glacier in the forward model (Figure 4.2(c)) and vice-versa. The discrepancies obtained with wrong rate factor, melting rate, and sliding constant can be larger than the mesh size. In the case of a perturbed target surface (Figure 5.25), the response in the bedrock topography attains a maximum of 15 m, whereas the maximal surface topography discrepancy is 40 m.

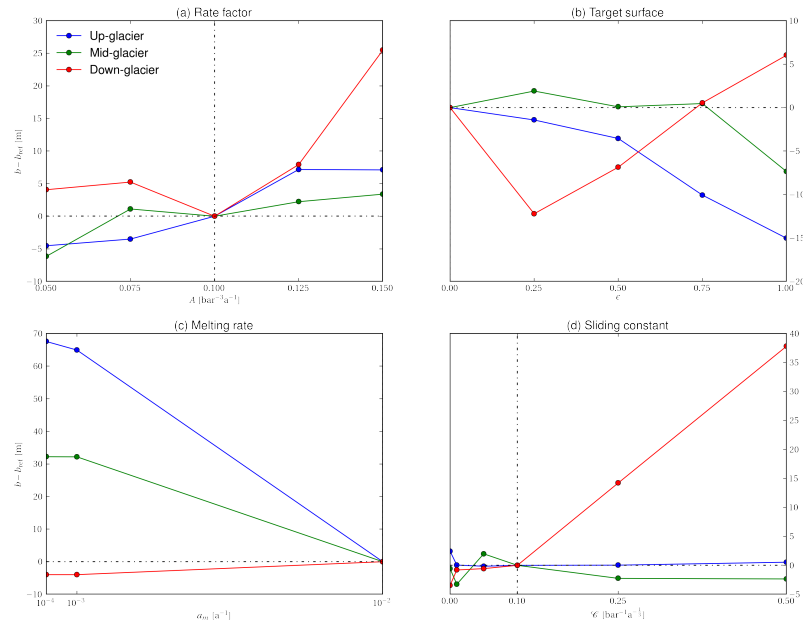


Figure 5.25 – Bedrock topography discrepancy with the reference topography b_{ref} obtained from the transient inversion with the right parameters as a function of the model parameters, indicated by the dashed lines. The results with varying sliding and equilibrium line altitudes z_{sl} and z_{ELA} are not represented here, since the transient inverse method is unable to make the surface misfit decrease in these cases. In (b), the ϵ is the parameter that defines the linear combination of s_i and s^{obs} given by (5.37).

5.7 Application to real-world data

This short section is devoted to the application of the quasi-stationary and transient inverse methods to real-world data. First, the shallow-ice-based quasi-stationary inverse method is applied to Gries and Silvretta glaciers. Then, the transient inverse method for Stokes ice flows is used on the two glaciers. In all cases, the surface mass-balance regularization parameter is set to $\varepsilon = 0.1$ and the rate factor to

$$A = 0.075686 \text{ bar}^{-3} \text{ a}^{-1}. \quad (5.38)$$

Then, on Silvretta glacier, the surface mass-balance parameters are set to [61]

$$\begin{aligned} z_{\text{ELA}} &= 2811 \text{ m} \\ m_1 = m_2 &= 0.0069 \text{ a}^{-1}. \end{aligned} \quad (5.39)$$

On Gries glacier, they are set to [61]

$$\begin{aligned} z_{\text{ELA}} &= 2937 \text{ m} \\ m_1 = m_2 &= 0.0088 \text{ a}^{-1}. \end{aligned} \quad (5.40)$$

Various sliding configurations are tested, especially on Gries glacier.

5.7.1 Quasi-stationary Inverse Method

The surface geometries of both Gries and Silvretta glaciers are inverted with Algorithm 4. The comparison of the estimated to the measured ice thickness is reported on Figure 5.26 under the assumption of no sliding. In view of the method's simplicity, the results are good and reasonable. The corresponding profile lines are displayed by Figures 5.27 and 5.28 for Silvretta and Gries glacier respectively. While the assumption of no sliding seems to be fine for Silvretta glacier, Figures 5.26 and 5.28 show significantly large defects in the estimated bedrock topography near the tongue of Gries glacier. Hence, Figure 5.28 also provides estimations for various sliding parameters, namely $z_{\text{sl}} = 2500 \text{ m}$ and \mathcal{C} varying between 0 and $1 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$. The value $\mathcal{C} = 0.5 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ seems to be the best compromise in the set of displayed values. The consideration of sliding basically results in the decrease of the ice thickness up-glacier and its increase down-glacier. The shape of the bedrock topography remains essentially the same in all cases.

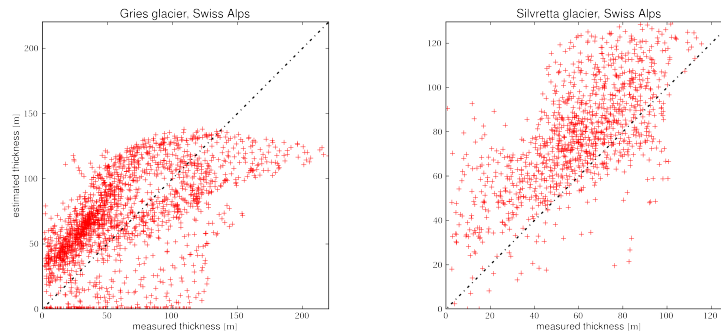


Figure 5.26 – Estimated ice thickness distribution (axis y) compared to the measured distribution (axis x) with the quasi-stationary inverse method (Algorithm 4) on Gries and Silvretta glaciers, Swiss Alps, under the assumption that no sliding applies. The surface mass-balance parameters applied are (5.39) and (5.40) for Silvretta and Gries glaciers respectively.

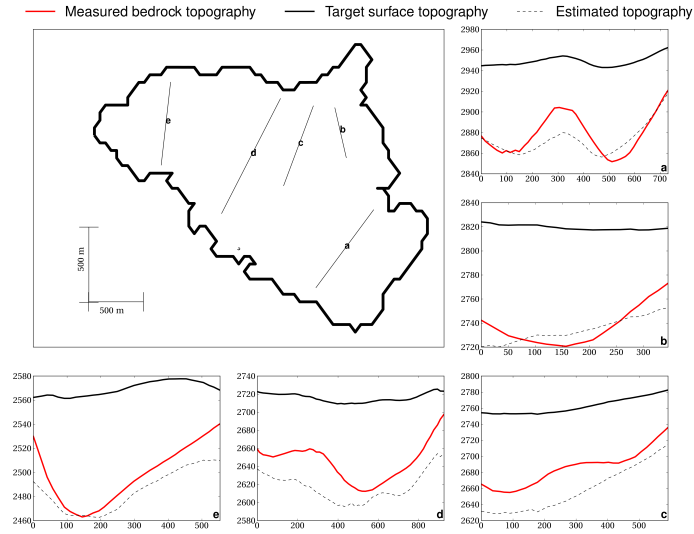


Figure 5.27 – Profile lines of the estimated bedrock topography (dashed curves) with the quasi-stationary inverse method compared to the measured topography (thick continuous red curves) on Silvretta glacier, under the assumption of no sliding. The estimation is good along all the depicted profile lines.

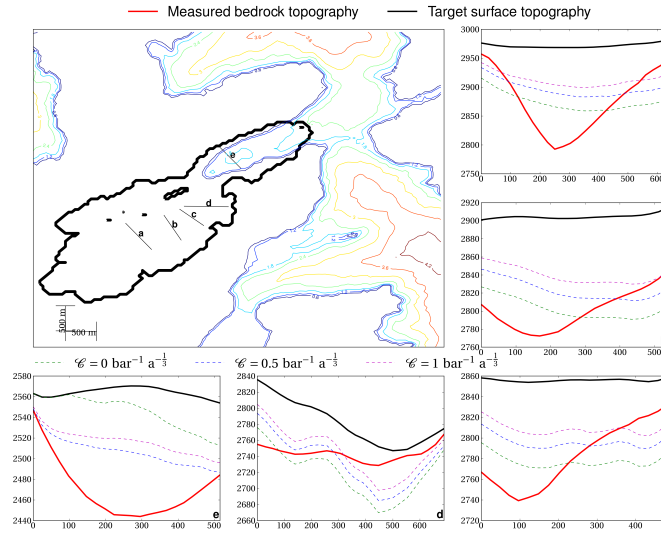


Figure 5.28 – Profile lines of the estimated bedrock topography (dashed curves) with the quasi-stationary inverse method compared to the measured topography (thick continuous red curves) on Gries glacier for various sliding configurations, namely $z_{sl} = 2500 \text{ m}$ and $\mathcal{C} = 0 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ (green, dashed curves), $\mathcal{C} = 0.5 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ (blue, dashed curves), and $\mathcal{C} = 1 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ (purple, dashed curves). The contour lines show the values of the sliding coefficient $\frac{1}{\mathcal{C}}$, expression (2.10), when $\mathcal{C} = 0.5 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$.

5.7.2 Transient Inverse Method

The transient inverse method is only applied in its Stokes version, as Alpine glaciers are closer to Stokes than shallow ice flows. However, the transient inverse method does not converge in this case, as explained previously. Moreover, the sliding and mass-balance parameters are subjected to measurement errors. Hence, it is very difficult to get a

good estimation of the bedrock topography. In addition, the estimation is mesh-dependent, and the finer the mesh, the more time-consuming the method. The coarser the mesh, the less precise the estimation. Figures 5.29 and 5.31 display the surface and bedrock topography misfits for various sliding configurations and the surface mass-balance parameters (5.39) and (5.40) for the geometries of Silvretta and Gries glaciers. Each colored curve corresponds to a different value of the sliding constant \mathcal{C} . The pictures show that the decrease of the misfit functions can be larger if the sliding law is chosen appropriately. However, Figures 5.30 and 5.32 show that the bedrock topography misfit ends up increasing, whatever the sliding parameters are. The final ice thickness estimation is not extraordinary in the end. The numerical experiments performed in this thesis show that the results strongly depend on the meshes used and the sensitivity analysis of Chapter 4 announced that the computation of the bedrock topography for Stokes ice flows is very difficult.

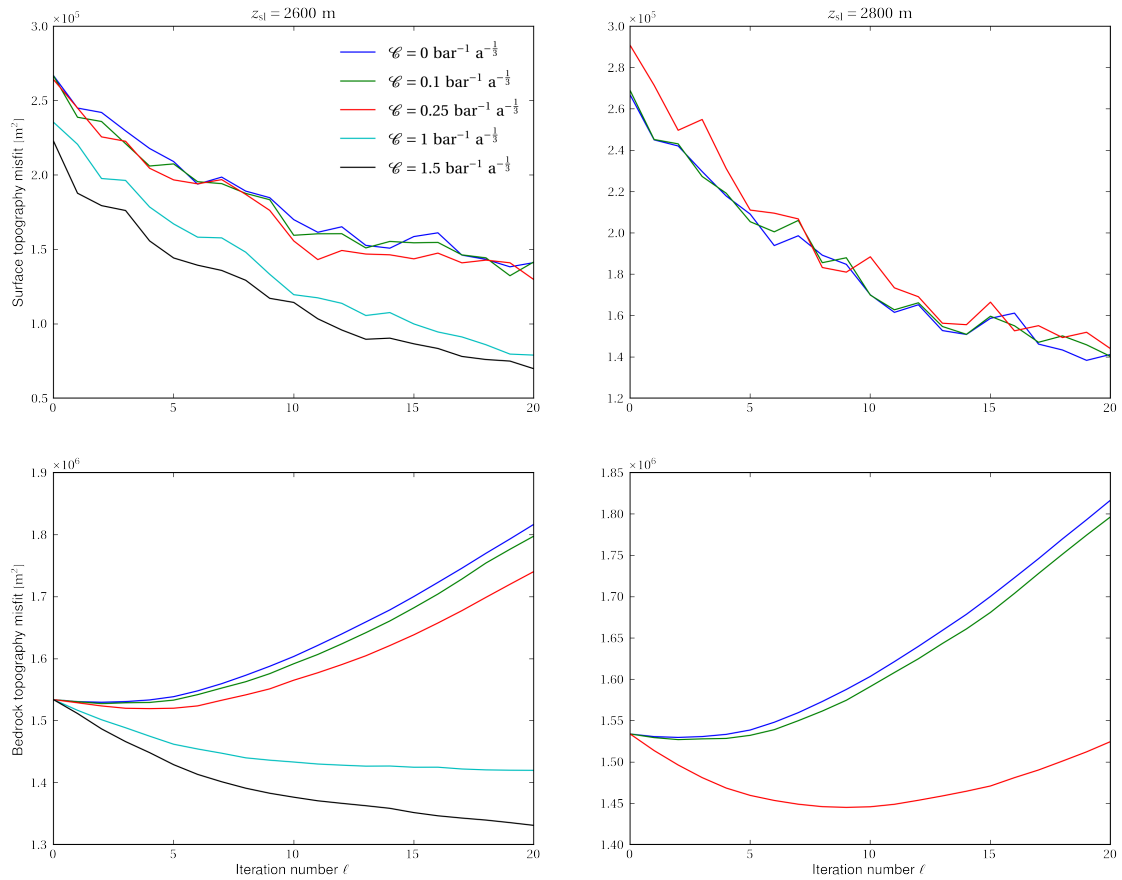


Figure 5.29 – Application of the transient inverse method (Algorithm 5) for a Stokes ice flow to Silvretta glacier, Swiss Alps, for various sliding parameters when the finite-elements' size is $H = 50$ m. The surface mass-balance parameters are (5.39). The best decrease occurs with $z_{sl} = 2600$ m and $\mathcal{C} = 1.5 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$.

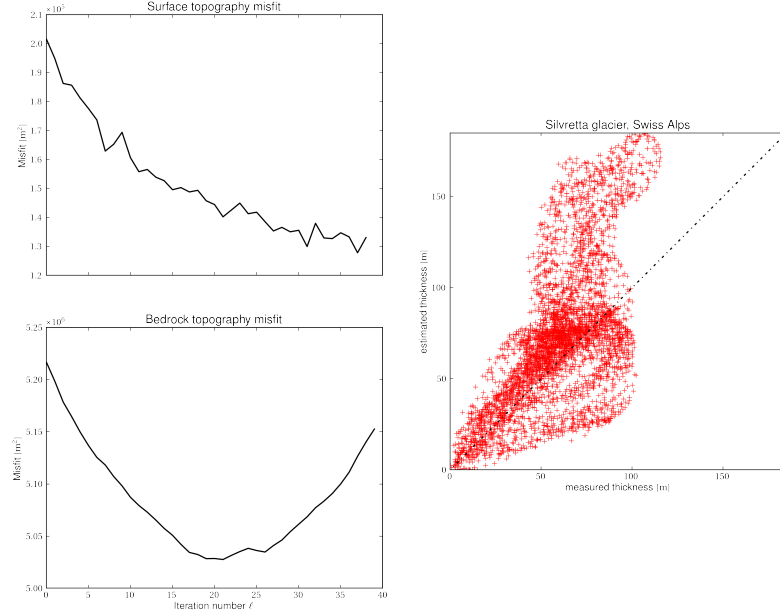


Figure 5.30 – Surface and bedrock topography misfits as functions of the iteration number (left) and resulting bedrock topography estimation after the 20-th iteration of the transient inverse method (Algorithm 5) for a Stokes ice flow on Silvretta glacier, Swiss Alps, when the finite-elements' size is $H = 25$ m, the surface mass-balance parameters are (5.39), and the sliding parameters are $z_{sl} = 2600$ m and $\mathcal{C} = 1.5 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ (right). The surface topography misfit decreases during the 40 displayed iterations, contrary to the bedrock topography misfit. The estimation displayed on the right is not as good as that obtained from the numerical validation that looks similar to the red one shown on Figure 5.22.

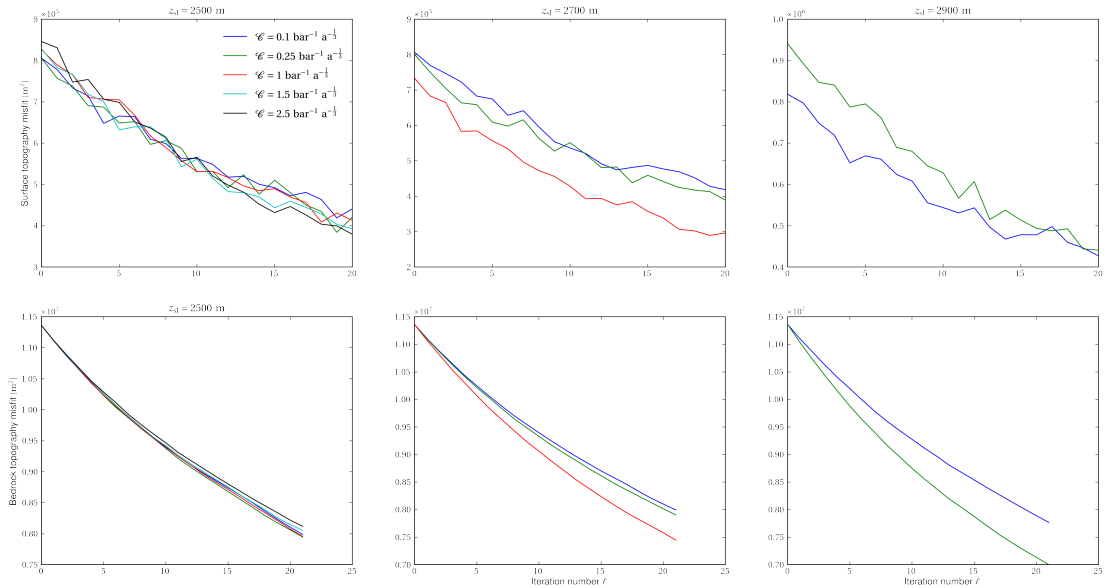


Figure 5.31 – Application of the transient inverse method (Algorithm 5) for a Stokes ice flow to Gries glacier, Swiss Alps, for various sliding parameters when the finite-elements' size is $H = 50$ m. The surface mass-balance parameters are (5.40). The best decrease occurs with $z_{sl} = 2900$ m and $\mathcal{C} = 0.25 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$.

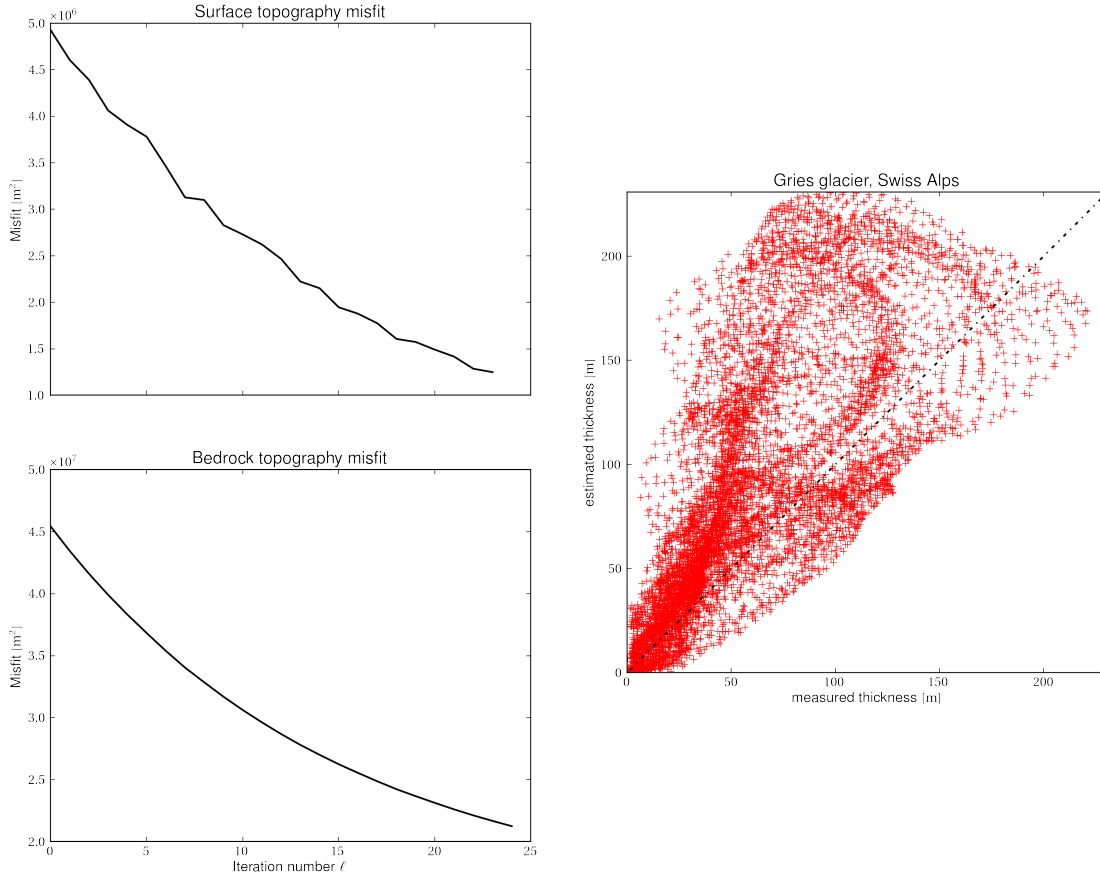


Figure 5.32 – Application of the transient inverse method (Algorithm 5) for a Stokes ice flow to Gries glacier, Swiss Alps, when the finite-elements' size is $H = 25$ m, the surface mass-balance parameters are (5.40), and the sliding parameters are $z_{sl} = 2900$ m and $\mathcal{C} = 0.25 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$.

5.8 Conclusion

This chapter introduces two methods to compute the subglacial topography from surface topography and mass-balance data, in two and three space dimensions. The quasi-stationary inverse method treats the transient problem as a stationary problem where the surface mass-balance is replaced with the apparent surface mass-balance defined as the balance between the surface topography evolution and the surface mass-balance. Then, the transient inverse method is formulated, that consists in the iterative update of the bedrock topography with the surface topography misfit. It is developed in both the shallow and Stokes ice approximations. Numerous numerical experiments are performed with the two-dimensional methods, as they are very cheap and the behavior of the three-dimensional methods are the same. Sensitivity analyses are exposed in the shallow ice approximation, that show the methods' susceptibilities to non-smooth data. In the Stokes ice approximation, the sensitivity of the transient inverse model to model parameters is shortly studied on the geometry of Silvretta glacier. Although the numerical validation is fine in all the presented cases, the application of the methods developed in this chapter to real-world data is not strikingly good. In particular, the transient inverse method for Stokes ice flows applied to real-world data does not supply as good results as desired. However, the quasi-stationary inverse method yields nice results in very little time, although it is based on the shallow ice approximation. The difficulties encountered by the inversion of Stokes

ice flows are not surprising, as stated in Chapter 4. The transient inverse method is certainly too basic for these computations and a shape optimization algorithm may be preferred. However, in this thesis, it is only developed for shallow ice flows.

6 Optimal control of the shallow ice model

The inversion methods presented in the previous chapter are not based on the gradient of the minimized cost functional. While they have the advantage of being easily implementable, they end up being slow, especially in three space dimensions. Furthermore, these methods are only capable of computing the subglacial topography, given the values of the model parameters.

Therefore, a faster, gradient-based method is now suggested. Its flexibility allows for the simultaneous computation of both the ice thickness distribution and some model parameters. Although its implementation requires much more coding and computations by hand than the previous methods, this versatile procedure has the convenience of being faster, more accurate, and more reliable. However, it is developed in the shallow ice approximation only, its extension to the Stokes ice approximation being shortly discussed by the end of this chapter and kept for a later project on the topic as it raises the consideration of further complications [1, 3], namely due to the coupling of the Stokes to the VOF equations. This first study highlights the possibilities of the method and improves targeting the search for new algorithms for the Stokes model.

A primal–dual method is advocated to solve the control problem. As before, the bedrock topography is discretized, and an estimation of its elevation is sought at every grid point, *i.e.* the bedrock topography is not parameterized. Hereafter, the optimal control method is applied to the discretized, shallow ice equations directly. This has the advantage of providing the exact gradient of the discrete objective function and a fully converging optimization process, at the expense of a longer time spent on coding. The amount of coding could be reduced with the use of forward automatic differentiation, at the expense of computational efficiency.

Contrary to the inverse methods of Chapter 5, the shape optimization method described in this chapter is able to take surface velocity data into account, even if they are only partially available on the glacier’s surface.

The chapter starts off formulating a simplified, two-dimensional shape optimization method. Next, its extension to three space dimensions is developed, from its most basic expression that only inverts for the bedrock topography to its most advanced formulation that infers some model parameters in the same time and accounts for possibly partial surface velocities and several surface topographies. Numerical results are then presented, before a short sensitivity analysis of the method and its application to real-world data. Finally, perspectives on the primal–dual method applied to the Stokes ice approximation are presented on continuous equations [1, 3], before global conclusions.

6.1 Two-dimensional description

Consider discretization (3.32) of problem (2.32) when the initial and final surfaces and the surface mass-balance are known but the bedrock geometry is unknown. More specifically, it is known that the glacier's surface topography evolves from s_i to s^{obs} from time t_i to time t_f (see Figure 6.1). Given this information, the purpose is to design a shape optimization algorithm (SOA) that reconstructs the subglacial topography b in two space dimensions. Its simplest expression is developed here that can be compared to the previously introduced two-dimensional methods (see Section 6.3.1). In particular, no sliding is assumed in this section and the surface mass-balance is constant with time.

Recall that the discretization of the space interval $\Omega_{\perp} = [x_1, x_2]$ is the grid $\tilde{\Omega}_{\perp}$ that consists of all the points $\xi_j = j\Delta x$, $0 \leq j \leq N_x$, where Δx is the grid spacing (see Figure 3.2). All the quantities are here expressed in vector form, whose components always refer to the value of the quantity at the position ξ_j , where j is the component's number. Time is discretized according to $t^{\ell} = t_i + \ell\Delta t$, where ℓ is a time index, $0 \leq \ell \leq M+1$, and Δt is the time step. In these notations, $t_f = t^{M+1}$.

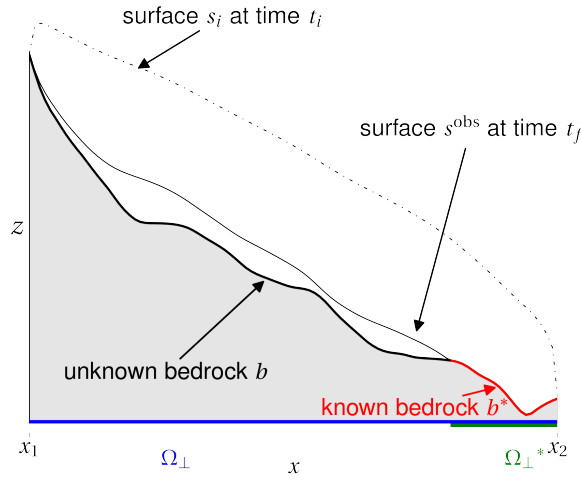


Figure 6.1 – Glacier in the Cartesian system of axes (x, z) . The surface s_i at time t_i evolves towards surface s^{obs} during the time interval $[t_i, t_f]$. The glacier's map domain Ω_{\perp} is indicated in blue and the subset Ω_{\perp}^* of Ω_{\perp} where the bedrock topography is known is marked in green.

The method consists in determining the optimal positions b_j , $0 \leq j \leq N_x$, of the bedrock topography such that the misfit between the computed and exact surfaces at the final time t_f is as small as possible. This is an optimal control problem where the ice thickness $\mathcal{H} = s - b$ is termed “state variable” and the bedrock topography b “control variable”. Recall the notations of section 3.3, where b , s , and \mathcal{H} are the vectors of components b_j , s_j , and \mathcal{H}_j respectively. The misfit functional to be minimized is

$$\mathcal{J}_h(\mathcal{H}, b) = \frac{1}{2} \|s_f(\mathcal{H}, b) - s^{\text{obs}}\|^2 + \mathcal{R}_h(b), \quad (6.1)$$

where $\|\cdot\|$ stands for the usual Euclidean norm, $s_f = b + \mathcal{H}^{M+1}$ is the computed surface at time $t_f = t^{M+1}$, and s^{obs} the vector of components $s^{\text{obs}}(\xi_j)$. The first term of this functional represents the misfit between the computed and the observed surface topographies and \mathcal{R}_h is a Tikhonov regularization term [112]. Following [63, 85], the

regularization

$$\mathcal{R}_h(\mathbf{b}) = \frac{r_1}{2} \sum_{j=0}^{N_x} \left| \frac{b_{j+1} - b_j}{\Delta x} \right|^2 \quad (6.2)$$

is advocated, so as to minimize oscillations in the bedrock topography. Now, the numerical scheme (3.32) links the control \mathbf{b} to the state variable $\mathcal{H}(\mathbf{b})$. The cost functional to be minimized can thus be expressed as

$$J_h(\mathbf{b}) = \mathcal{J}_h(\mathcal{H}(\mathbf{b}), \mathbf{b}). \quad (6.3)$$

The minimization problem is cast into: *find*

$$\underset{\mathbf{b} \in \mathcal{U}_h^{\text{ad}}}{\operatorname{argmin}} J_h(\mathbf{b}) \quad (6.4)$$

under the constraint that the ice thickness and the bedrock topography satisfy equation (3.32), where $\mathcal{U}_h^{\text{ad}}$ is the set of admissible solutions described later in this section. This constrained minimization problem is here reformulated as the unconstrained minimization of the Lagrangian functional

$$\mathcal{L}_h(\mathcal{H}, \boldsymbol{\lambda}, \mathbf{b}) = \mathcal{J}_h(\mathcal{H}, \mathbf{b}) + \left[\mathcal{H}^0 - \mathbf{s}_i + \mathbf{b} \right]^T \boldsymbol{\lambda}^0 + \sum_{\ell=0}^M \Delta t \left[\frac{\mathcal{H}^{\ell+1} - \mathcal{H}^\ell}{\Delta t} - \mathcal{F}^\ell - \mathcal{B}^\ell \right]^T \boldsymbol{\lambda}^{\ell+1}, \quad (6.5)$$

where \mathcal{F} is the vector of components \mathcal{F}_j^ℓ and $\boldsymbol{\lambda}^\ell$ the costate variable of \mathcal{H}^ℓ , *i.e.* the Lagrange multiplier of constraint (3.32). Note that it is not necessary to take the boundary condition into account, as it is implicitly understood in the constraint equation. The solutions of the minimization problem are among the stationary points of \mathcal{L}_h that satisfy the optimality conditions [9, 52, 40, 118]

$$\nabla \mathcal{L}_h(\mathcal{H}, \boldsymbol{\lambda}, \mathbf{b}) = 0. \quad (6.6)$$

The stationarity of \mathcal{L}_h with respect to the costate variable $\boldsymbol{\lambda}$ yields the constraint equation (3.32). The stationarity of \mathcal{L}_h with respect to the state variable \mathcal{H} corresponds to the dual problem

$$-\frac{\lambda_j^{\ell+1} - \lambda_j^\ell}{\Delta t} = \tilde{\mathcal{F}}_j^\ell, \quad 1 \leq \ell \leq M, \quad (6.7)$$

with vanishing boundary conditions and a scheme-dependent final condition expressed below. This is the discretization of the backward diffusion-transport equation

$$\begin{cases} -\frac{\partial \lambda}{\partial t} + (n+2)\mathcal{F}_f \frac{\partial}{\partial x}(b + \mathcal{H}) \frac{\partial \lambda}{\partial x} - n \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial \lambda}{\partial x} \right) = 0, & \text{in } \Omega_\perp, \quad t_i \leq t \leq t_f, \\ \lambda = s^{\text{obs}} - s_f, & \text{in } \Omega_\perp, \quad t = t_f, \\ \lambda = 0, & \text{on } \partial\Omega_\perp, \quad t_i \leq t \leq t_f, \end{cases} \quad (6.8)$$

where

$$\mathcal{F}_f(b, \mathcal{H}) = \Gamma \mathcal{H}^{n+1} \left| \frac{\partial}{\partial x}(b + \mathcal{H}) \right|^{n-1}. \quad (6.9)$$

The right-hand side of (6.7) writes, in the semi-implicit time discretization,

$$\tilde{\mathcal{F}}_j^\ell = \frac{1}{\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^{\ell-1} \frac{\lambda_{j+1}^\ell - \lambda_j^\ell}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^{\ell-1} \frac{\lambda_j^\ell - \lambda_{j-1}^\ell}{\Delta x} \right]$$

$$\begin{aligned}
 & + \frac{n-1}{\Delta x} \left[(\mathcal{G}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} - (\mathcal{G}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right] \\
 & - \frac{n+2}{2} \left[(\mathcal{F}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} + (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right],
 \end{aligned} \tag{6.10}$$

where $(\mathcal{F}_f)_{j-\frac{1}{2}}^\ell$ is the discrete counterpart

$$(\mathcal{F}_f)_{j-\frac{1}{2}}^\ell = \Gamma \left(\frac{\mathcal{H}_j^\ell + \mathcal{H}_{j-1}^\ell}{2} \right)^{n+1} \left| \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} \right|^{n-1} \tag{6.11}$$

of (6.9). The corresponding final condition is

$$\lambda_j^{M+1} - \frac{\Delta t}{\Delta x} \left((\mathcal{D}_f)_{j+\frac{1}{2}}^M \frac{\lambda_{j+1}^{M+1} - \lambda_j^{M+1}}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^M \frac{\lambda_j^{M+1} - \lambda_{j-1}^{M+1}}{\Delta x} \right) = s^{\text{obs}}(\xi_j) - (s_f)_j^{M+1}. \tag{6.12}$$

This is solved with a scheme similar to that solving (6.7) (see Appendix B.1). Moreover, due to the Lagrangian's structure, the components of λ^0 are found directly with the expression

$$\begin{aligned}
 \lambda_j^0 &= \lambda_j^1 - (n+2) \frac{\Delta t}{2} \left((\mathcal{F}_f)_{j+\frac{1}{2}}^0 \frac{s_{j+1}^1 - s_j^1}{\Delta x} \frac{\lambda_{j+1}^1 - \lambda_j^1}{\Delta x} + (\mathcal{F}_f)_{j-\frac{1}{2}}^0 \frac{s_j^1 - s_{j-1}^1}{\Delta x} \frac{\lambda_j^1 - \lambda_{j-1}^1}{\Delta x} \right) \\
 & + (n-1) \frac{\Delta t}{\Delta x} \left((\mathcal{G}_f)_{j+\frac{1}{2}}^0 \frac{s_{j+1}^1 - s_j^1}{\Delta x} \frac{\lambda_{j+1}^1 - \lambda_j^1}{\Delta x} - (\mathcal{G}_f)_{j-\frac{1}{2}}^0 \frac{s_j^1 - s_{j-1}^1}{\Delta x} \frac{\lambda_j^1 - \lambda_{j-1}^1}{\Delta x} \right),
 \end{aligned} \tag{6.13}$$

where \mathcal{D}_f and \mathcal{F}_f are defined by (3.34) and (6.11) and

$$(\mathcal{G}_f)_{j-\frac{1}{2}}^\ell = \left(\frac{\mathcal{H}_j^\ell + \mathcal{H}_{j-1}^\ell}{\Delta x} \right)^{n+2} \left| \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} \right|^{n-3} \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x}. \tag{6.14}$$

instead of solving the system of equations (6.7) with right-hand side (6.10).

In the implicit time discretization, the right-hand side of the dual problem (6.7) becomes

$$\begin{aligned}
 \tilde{\mathcal{F}}_j^\ell &= -\frac{(n+2)}{4} \left[(\mathcal{F}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} \left(\frac{\lambda_{j+1}^\ell - \lambda_j^\ell}{\Delta x} + \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} \right) + (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} \left(\frac{\lambda_j^\ell - \lambda_{j-1}^\ell}{\Delta x} + \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right) \right] \\
 & + \frac{n}{2\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \left(\frac{\lambda_{j+1}^\ell - \lambda_j^\ell}{\Delta x} + \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} \right) - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \left(\frac{\lambda_j^\ell - \lambda_{j-1}^\ell}{\Delta x} + \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right) \right],
 \end{aligned} \tag{6.15}$$

The corresponding final condition writes

$$\begin{aligned}
 \lambda_j^{M+1} &+ (n+2) \frac{\Delta t}{4} \left((\mathcal{F}_f)_{j+\frac{1}{2}}^{M+1} \frac{s_{j+1}^{M+1} - s_j^{M+1}}{\Delta x} \frac{\lambda_{j+1}^{M+1} - \lambda_j^{M+1}}{\Delta x} + (\mathcal{F}_f)_{j-\frac{1}{2}}^{M+1} \frac{s_j^{M+1} - s_{j-1}^{M+1}}{\Delta x} \frac{\lambda_j^{M+1} - \lambda_{j-1}^{M+1}}{\Delta x} \right) \\
 & - n \frac{\Delta t}{2\Delta x} \left((\mathcal{D}_f)_{j+\frac{1}{2}}^{M+1} \frac{\lambda_{j+1}^{M+1} - \lambda_j^{M+1}}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^{M+1} \frac{\lambda_j^{M+1} - \lambda_{j-1}^{M+1}}{\Delta x} \right) = s^{\text{obs}}(\xi_j) - (s_f)_j.
 \end{aligned} \tag{6.16}$$

This is solved with a scheme similar to that solving (6.7) (see Appendix B.1). Moreover, due to the Lagrangian's

structure, the components of λ^0 are found directly with the expression

$$\begin{aligned} \lambda_j^0 = & \lambda_j^1 - (n+2) \frac{\Delta t}{4} \left((\mathcal{F}_f)^0_{j+\frac{1}{2}} \frac{s_{j+1}^0 - s_j^0}{\Delta x} \frac{\lambda_{j+1}^1 - \lambda_j^1}{\Delta x} + (\mathcal{F}_f)^0_{j-\frac{1}{2}} \frac{s_j^0 - s_{j-1}^0}{\Delta x} \frac{\lambda_j^1 - \lambda_{j-1}^1}{\Delta x} \right) \\ & + n \frac{\Delta t}{2\Delta x} \left((\mathcal{D}_f)^0_{j+\frac{1}{2}} \frac{\lambda_{j+1}^1 - \lambda_j^1}{\Delta x} - (\mathcal{D}_f)^0_{j-\frac{1}{2}} \frac{\lambda_j^1 - \lambda_{j-1}^1}{\Delta x} \right), \end{aligned} \quad (6.17)$$

instead of solving the system of equations (6.7) with right-hand side (6.15).

Finally, the stationarity of \mathcal{L}_h with respect to the control \mathbf{b} gives

$$\begin{aligned} \frac{\partial \mathcal{L}_h}{\partial b_j} = & -(n-1) \frac{\Delta t}{\Delta x} \sum_{\ell=0}^M \left((\mathcal{G}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} - (\mathcal{G}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right) \\ & - \frac{\Delta t}{\Delta x} \sum_{\ell=0}^M \left((\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right) + (s_f)_j - s^{\text{obs}}(\xi_j) + \lambda_j^0 + \frac{\partial \mathcal{R}_h}{\partial b_j} \end{aligned} \quad (6.18)$$

in the semi-implicit time discretization, and

$$\begin{aligned} \frac{\partial \mathcal{L}_h}{\partial b_j} = & -\frac{n\Delta t}{2\Delta x} \sum_{\ell=0}^M \left[\left((\mathcal{D}_f)_{j+\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j+\frac{1}{2}}^{\ell+1} \right) \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} - \left((\mathcal{D}_f)_{j-\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j-\frac{1}{2}}^{\ell+1} \right) \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right] \\ & + (s_f)_j - s^{\text{obs}}(\xi_j) + \lambda_j^0 + \frac{\partial \mathcal{R}_h}{\partial b_j} \end{aligned} \quad (6.19)$$

in the implicit time discretization. Although these expressions seem very different, they are consistent with each other. The terms proportional to \mathcal{G} in (6.18) arise because of the mixing of the diffusivities at time iteration ℓ and the surface topography gradients at time iteration $\ell+1$ in the forward scheme. Formally, $\mathcal{G} \|\nabla s\|$ corresponds to \mathcal{D} . In both cases, the derivative of the regularization term amounts to

$$\frac{\partial \mathcal{R}_h}{\partial b_j} = \frac{r_1}{\Delta x^2} (2b_j - b_{j-1} - b_{j+1}). \quad (6.20)$$

These gradients correspond to the discretizations of

$$\frac{\partial \mathcal{L}}{\partial b} \hat{b} = -n \int_{t_i}^{t_f} \int_{x_1}^{x_2} \frac{\partial}{\partial x} \left(\mathcal{D}_f \frac{\partial \lambda}{\partial x} \right) \hat{b} \, dx \, dt + \int_{x_1}^{x_2} \left(s_f - s^{\text{obs}} + \lambda \Big|_{t=t_i} \right) \hat{b} \, dx + \frac{\partial \mathcal{R}}{\partial b} \hat{b}, \quad (6.21)$$

for all functions \hat{b} .

Now, when the state and costate variables satisfy the constraint equation (3.32) and the dual problem (6.7), the Lagrangian (6.5) is precisely the cost functional (6.1), namely

$$\mathcal{L}_h(\mathcal{H}(\mathbf{b}), \lambda(\mathbf{b}), \mathbf{b}) = \mathcal{J}_h(\mathcal{H}(\mathbf{b}), \mathbf{b}) = J_h(\mathbf{b}). \quad (6.22)$$

The dual problem links the control \mathbf{b} to the costate variable $\lambda(\mathbf{b})$. Because of (3.32) and (6.7), the derivative of (6.22) with respect to \mathbf{b} is, by the chain rule,

$$\frac{dJ_h}{db_j} = \frac{\partial \mathcal{L}_h}{\partial b_j}(\mathcal{H}(\mathbf{b}), \lambda(\mathbf{b}), \mathbf{b}). \quad (6.23)$$

This method relies on the so-called backward sensitivity of J_h with respect to the bedrock topography. The minimization problem can also be solved with the cost functional's forward sensitivity by means of an automatic differentiation algorithm. Although such a procedure is not advocated here because it is much slower for a shape

optimization problem due to the large amount of control parameters, it is shortly explained in Section 6.2.4.

Up to this point, no information was supplied about admissible solutions to the minimization problem (6.4) under constraint (3.32). In general, the glacier's boundaries may change with time. In particular, it may be that part of the bedrock topography inside domain $\tilde{\Omega}_\perp$ is known (*e.g.* the red part of the bedrock topography on Figure 6.1). Let $\tilde{\Omega}_\perp^* \subset \tilde{\Omega}_\perp$ be the domain where the bedrock topography is known (green part of the map domain on Figure 6.1). At these locations, it is known because the ice thickness is zero during a period of time in the evolution from time t_i to time t_f and is denoted by b^* :

$$b_j = b^*(\xi_j), \quad \text{in } \tilde{\Omega}_\perp^*. \quad (6.24)$$

Recall that the tilded space $\tilde{\Omega}_\perp^*$ refers to the discretized grid of the glacier's map domain Ω_\perp^* (see Figure 3.2).

Additionally, in the case of retreating glaciers, since the source term \mathcal{B}_j is negative below the equilibrium line altitude, the solution topography b_j to the minimization problem must satisfy

$$b_j < \min\{s_i(\xi_j), s^{\text{obs}}(\xi_j)\}, \quad \text{in } \tilde{\Omega}_\perp \setminus \tilde{\Omega}_\perp^*, \quad (6.25)$$

to make the method applicable. Note the strict inequality. It is important to force b_j to be strictly below the glacier's surface topography, because the algorithm cannot recover the bedrock topography where $\mathcal{B} < 0$ if it has already updated its altitude in such a way that the thickness vanishes. If (6.25) is not imposed, then the problem is ill-posed, as the solution $\mathcal{H} = 0$ also provides a zero misfit functional, since the surface mass-balance is applied only at the locations where $\mathcal{H} > 0$. The space of admissible control functions is therefore

$$\mathcal{U}_h^{\text{ad}} = \left\{ \mathbf{b} \in \mathbb{R}^{N_x} : b_j = b^*(\xi_j) \text{ in } \tilde{\Omega}_\perp^* \text{ and } b_j < \min\{s_i(\xi_j), s^{\text{obs}}(\xi_j)\} \text{ in } \tilde{\Omega}_\perp \setminus \tilde{\Omega}_\perp^* \right\}. \quad (6.26)$$

Finally, in order to minimize the cost functional (6.3), the solution of the equation

$$\frac{dJ_h}{d\mathbf{b}}(\mathbf{b}) = 0 \quad (6.27)$$

is sought in the space $\mathcal{U}_h^{\text{ad}}$, which is realized by iteratively solving (6.27) with a quasi-Newton method while projecting the control variable onto $\mathcal{U}_h^{\text{ad}}$, as stated in Algorithm 6, where the Hessian matrix of J_h is approximated with the BFGS method [15, 35, 47, 107]. The Algorithm is run until stopping criterion

$$\|\mathbf{b}^{m+1} - \mathbf{b}^m\| = \max_{0 \leq j \leq N_x} |b_j^{m+1}(x) - b_j^m(x)| < \delta \quad (6.28)$$

is met, for a real constant tolerance $\delta > 0$, where \mathbf{b}^m is the bedrock estimation at iteration m . Other stopping criteria are more relevant for this problem. However, criterion (6.28) is chosen so as to offer the possibility to compare the SOA to the methods introduced in the previous chapters. All the relevant implementation details are given in Appendix B.1.

Note that the Lagrangian (6.5) reduces to

$$\mathcal{L}_h(\mathcal{H}, \boldsymbol{\lambda}, \mathbf{b}) = \mathcal{J}_h(\mathcal{H}, \mathbf{b}) + \boldsymbol{\mathcal{T}}^T \boldsymbol{\lambda}, \quad (6.29)$$

when the superscript ℓ is dropped. One may want to deduce the steady counterparts of (6.7) and corresponding gradient. Such a procedure is not advised for steady geometries because it is time-consuming and lacks robustness. Indeed, it is not clear whether a stationary state exists for both the primal and dual problems, starting from any bedrock topography \mathbf{b}^m and initial surface \mathbf{s}_i . Moreover, its computation may take a large amount of time, especially during the first BFGS iterations. In practice, the steady problem is rather solved time-dependently, by setting

Algorithm 6 Two-dimensional shape optimization algorithm

set glacier geometry with initial bedrock \mathbf{b}^0 and surface \mathbf{s}_i
while $\|\mathbf{b}^{m+1} - \mathbf{b}^m\| > \delta$ **do**
 solve the primal problem (3.32)
 solve the dual problem (6.7) with right-hand side (6.10) in the semi-implicit time discretization or (6.15) in the implicit time discretization
 compute the gradient (6.18) (semi-implicit time discretization) or (6.19) (implicit time discretization)
 compute

$$\mathbf{b}^{m+1} = \mathbf{b}^m + \left(\frac{d^2 J_h}{d\mathbf{b}^2}(\mathbf{b}^m) \right)^{-1} \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}^m),$$

 where the Hessian is approximated with the BFGS method
 project the vector \mathbf{b}^{m+1} onto the space of admissible controls (6.26).
end while

$\mathbf{s}_i = \mathbf{s}^{\text{obs}}$ and choosing a large enough time frame $[t_i, t_f]$. Instead of computing the stationary state at every iteration, the evolution of the surface topography is computed over the time frame $[t_i, t_f]$ and its deviation from the true stationary geometry is studied. Because of stationarity, such a deviation must be zero when the algorithm has converged. This procedure saves significant computational effort compared to a purely stationary algorithm.

The precision of the gradient is crucial for the optimization algorithm. To check it, Taylor's expansion of J_h

$$J_h(\mathbf{b} + \varepsilon \hat{\mathbf{b}}) = J_h(\mathbf{b}) + \varepsilon \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}) \cdot \hat{\mathbf{b}} + \mathcal{O}(\varepsilon^2), \quad (6.30)$$

is considered [22]. It must satisfy

$$\lim_{\varepsilon \rightarrow 0} \frac{J_h(\mathbf{b} + \varepsilon \hat{\mathbf{b}}) - J_h(\mathbf{b})}{\varepsilon \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}) \cdot \hat{\mathbf{b}}} = 1. \quad (6.31)$$

Numerical experiments with some reasonable \mathbf{b} and the normalized vector

$$\hat{\mathbf{b}} = \frac{\frac{dJ_h}{d\mathbf{b}}(\mathbf{b})}{\left\| \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}) \right\|}. \quad (6.32)$$

give results that verify relation (6.31). Hence the implementation of the discretizations presented in this section can be trusted.

6.2 Three-dimensional formulation

In this section, the direct, three-dimensional extension of the shape optimization algorithm introduced in the previous section is formulated. It computes the bedrock topography from various field observations. Sliding is considered along with a time-dependent surface mass-balance (2.34). It is improved to incorporate more data and infer the value of the model parameters. For the sake of clarity and readability, the whole section is devoted to the development of the method step-by-step, because the derivations lead to quite complicated formulae.

6.2.1 Basic algorithm

Consider discretization (3.14)–(3.18) of problem (2.31) when the initial and final surfaces and the surface mass-balance are known but the bedrock geometry is unknown. As in the previous section, it is known that the glacier's

surface topography evolves from s_i to s^{obs} from time t_i to time t_f (see Figure 6.1). Given this information, the purpose is to design a shape optimization algorithm (SOA) that reconstructs the subglacial topography b in three space dimensions. This time, sliding is taken into account and the surface mass-balance is assumed time-dependent.

Recall that the discretization of the map domain $\Omega_{\perp} = [0, L_x] \times [0, L_y]$ is the grid $\tilde{\Omega}_{\perp}$ that consists of all the points $(\xi_j, \zeta_k) = (j, k)\Delta x$, $0 \leq j \leq N_x$, $0 \leq k \leq N_y$, where Δx is the grid spacing. All the quantities are here expressed in vector form, whose components always refer to the value of the quantity at the position (ξ_j, ζ_k) . Time is discretized according to $t^{\ell} = t_i + \ell\Delta t$, where ℓ is a time index, $0 \leq \ell \leq M+1$, and Δt is the time step.

The purpose of the basic algorithm is to determine such a subglacial topography \mathbf{b} that the cost functional

$$\mathcal{J}_h(\mathcal{H}, \mathbf{b}) = \frac{1}{2} \left\| s_f(\mathcal{H}, \mathbf{b}) - s^{\text{obs}} \right\|^2 + \mathcal{R}_h(\mathbf{b}) \quad (6.33)$$

is as small as possible, under the constraint that \mathcal{H} satisfies the evolution scheme (3.14)-(3.15)-(3.16)-(3.18). The regularization is this time the following combination of minimal potential and curvature

$$\mathcal{R}_h(\mathbf{b}) = \frac{r_1}{2} \sum_{j=1}^{N_x-1} \sum_{k=1}^{N_y-1} \left[\left(\frac{b_{j+1,k} - b_{jk}}{\Delta x} \right)^2 + \left(\frac{b_{j,k+1} - b_{jk}}{\Delta x} \right)^2 \right] + \frac{r_2}{2} \sum_{j=2}^{N_x-2} \sum_{k=2}^{N_y-2} \left(\frac{b_{j+1,k} + b_{j,k+1} + b_{j-1,k} + b_{j,k-1} - 4b_{ij}}{\Delta x^2} \right)^2, \quad (6.34)$$

where $r_i \geq 0$, $i = 1, 2$, are two regularization constants. The first term corresponds to the squared norm of the bedrock topography's gradient, while the second term to that of its Laplacian. The space of admissible solutions is

$$\mathcal{U}_h^{\text{ad}} = \left\{ \mathbf{b} \in \mathbb{R}^{N_x \times N_y} : b_{jk} = b_{jk}^* \text{ in } \tilde{\Omega}_{\perp}^* \text{ and } b_{jk} < \min\{s_i(\xi_j, \zeta_k), s^{\text{obs}}(\xi_j, \zeta_k)\} \text{ in } \tilde{\Omega}_{\perp} \setminus \tilde{\Omega}_{\perp}^* \right\}, \quad (6.35)$$

where $\tilde{\Omega}_{\perp}$ and $\tilde{\Omega}_{\perp}^*$ are the discrete grids of Ω_{\perp} and Ω_{\perp}^* respectively. The Lagrangian associated to this problem is

$$\begin{aligned} \mathcal{L}_h(\mathcal{H}, \overline{\mathcal{H}}, \overline{\mathcal{H}}, \overline{\lambda}, \overline{\lambda}, \overline{\lambda}, \mathbf{b}) &= \mathcal{J}_h(\mathcal{H}, \mathbf{b}) + \left[\mathcal{H}^0 - s_i + \mathbf{b} \right]^T \boldsymbol{\lambda}^0 + \sum_{\ell=0}^M \Delta t \left[\frac{\overline{\mathcal{H}}^{\ell} - \mathcal{H}^{\ell}}{\Delta t} - \mathcal{F}^{\ell} \right]^T \overline{\boldsymbol{\lambda}}^{\ell} \\ &+ \sum_{\ell=0}^M \Delta t \left[\frac{\overline{\mathcal{H}}^{\ell} - \mathcal{H}^{\ell}}{\Delta t} - \vartheta(\overline{\mathcal{H}}^{\ell}) \overline{\mathcal{B}}^{\ell} \right]^T \overline{\boldsymbol{\lambda}}^{\ell} + \sum_{\ell=0}^M \left[\mathcal{H}^{\ell+1} - [\overline{\mathcal{H}}^{\ell}]^+ \right]^T \boldsymbol{\lambda}^{\ell+1}, \end{aligned} \quad (6.36)$$

where $\overline{\mathcal{B}}^{\ell}$ is the surface mass-balance evaluated in $\overline{\mathcal{H}}^{\ell}$, $\boldsymbol{\lambda}^{\ell}$, $\overline{\boldsymbol{\lambda}}^{\ell}$, and $\overline{\boldsymbol{\lambda}}^{\ell}$ are the costate variables of \mathcal{H}^{ℓ} , $\overline{\mathcal{H}}^{\ell}$, and $\overline{\mathcal{H}}^{\ell}$, i.e. the Lagrange multipliers of constraints (3.14), (3.15), (3.16), and (3.18). The condition $\mathcal{H}_{jk} \geq 0$ is now taken explicitly into account in the last term of Lagrangian (6.36). Note that this Lagrangian function is significantly more complicated than its two-dimensional counterpart (6.5), because the evolution equation is now splitted into three separated steps. Moreover, sliding is here considered and is part of vector \mathcal{F}^{ℓ} . The solutions of the minimization problem are among the stationary points of \mathcal{L}_h that satisfy the optimality condition [9, 52, 40, 118]

$$\nabla \mathcal{L}_h(\mathcal{H}, \overline{\mathcal{H}}, \overline{\mathcal{H}}, \overline{\lambda}, \overline{\lambda}, \overline{\lambda}, \mathbf{b}) = 0. \quad (6.37)$$

In this simple, three-dimensional case, the dual problem is solved in the following way: Finalize the dual variable with

$$\lambda_{jk}^{M+1} = s^{\text{obs}}(\xi_j, \zeta_k) - (s_f)_{jk}. \quad (6.38)$$

Then, for $\ell = M, M-1, \dots, 1, 0$, first solve

$$\sum_{il} \left(\delta_{ij} \delta_{kl} - \Delta t \vartheta(\overline{\mathcal{H}}_{il}^\ell) (\nabla_{\mathcal{H}\overline{\mathcal{B}}}^\ell)_{il;jk}^\ell \right) \overline{\lambda}_{il}^\ell = \vartheta(\overline{\mathcal{H}}_{jk}^\ell) \lambda_{jk}^{\ell+1}, \quad (6.39)$$

where ϑ is the Heaviside function (3.17); second, solve the discretized linear, backward advection-diffusion equation

$$-\frac{\overline{\lambda}_{jk}^\ell - \overline{\lambda}_{jk}^\ell}{\Delta t} = (\tilde{\mathcal{T}}_1)_{jk}^\ell; \quad (6.40)$$

finally, solve

$$-\frac{\overline{\lambda}_{jk}^\ell - \lambda_{jk}^\ell}{\Delta t} = (\tilde{\mathcal{T}}_2)_{jk}^\ell. \quad (6.41)$$

This is the discretization of the continuous, backward diffusion-transport equation

$$-\frac{\partial \lambda}{\partial t} + \left((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s \right) \nabla s \cdot \nabla \lambda - (n-1) \nabla \cdot \left(\mathcal{G} \nabla s \cdot \nabla \lambda \nabla s \right) - \nabla \cdot \left(\mathcal{D} \nabla \lambda \right) = 0, \quad \text{in } \Omega_\perp \times [t_i, t_f] \quad (6.42)$$

where

$$\mathcal{F}_f = \Gamma \mathcal{H}^{n+1} \left\| \nabla s \right\|^{n-1} \quad (6.43)$$

$$\mathcal{F}_s = \Gamma_s \mathcal{H}^n \left\| \nabla s \right\|^{n-1} \quad (6.44)$$

$$\mathcal{G} = \left(\Gamma \mathcal{H} + \Gamma_s \right) \mathcal{H}^{n+1} \left\| \nabla s \right\|^{n-3} \quad (6.45)$$

are the continuous counterparts of (A.30) and (A.32). Here, a few remarks are necessary. First, no linear system needs to be built in order to solve (6.41), since $(\tilde{\mathcal{T}}_2)_{jk}^\ell$ is independent of λ_{jk}^ℓ . The precise expressions of $(\tilde{\mathcal{T}}_1)_{jk}^\ell$ and $(\tilde{\mathcal{T}}_2)_{jk}^\ell$ are given below because they depend on the time discretization. Second, the right-hand side of (6.40) is, strictly speaking,

$$(\tilde{\mathcal{T}}_1)_{jk}^\ell + \delta(\overline{\mathcal{H}}_{jk}^\ell) \overline{\mathcal{B}}_{jk}^\ell \overline{\lambda}_{jk}^\ell. \quad (6.46)$$

However, the second term of this expression simplifies to zero because if $\overline{\mathcal{H}}_{jk}^\ell = 0$, then $\overline{\mathcal{H}}_{jk}^\ell = 0$ from (3.16), whence $\overline{\lambda}_{jk}^\ell = 0$ from (6.39). Finally, note that if the surface mass-balance is defined by (2.34), then $\nabla_{\mathcal{H}\overline{\mathcal{B}}}^\ell$ in (6.39) is the diagonal matrix of components

$$(\nabla_{\mathcal{H}\overline{\mathcal{B}}}^\ell)_{il;jk}^\ell = \delta_{ij} \delta_{kl} \left(\frac{m_1 + m_2}{2} - \left(\frac{m_1 - m_2}{2} \right)^2 \frac{b_{jk} + \overline{\mathcal{H}}_{jk}^\ell - z_{\text{ELA}}}{\sqrt{\left(\frac{m_1 - m_2}{2} (b_{jk} + \overline{\mathcal{H}}_{jk}^\ell - z_{\text{ELA}}) \right)^2 + \varepsilon}} \right). \quad (6.47)$$

Write

$$\left(\delta_x^c s \right)_{j-\frac{1}{2}, k-\frac{1}{2}} = \frac{1}{2} \left(\frac{s_{j,k-1}^\ell - s_{j-1,k-1}^\ell}{\Delta x} + \frac{s_{jk}^\ell - s_{j-1,k}^\ell}{\Delta x} \right) \quad (6.48)$$

$$\left(\delta_y^c s \right)_{j-\frac{1}{2}, k-\frac{1}{2}} = \frac{1}{2} \left(\frac{s_{j-1,k}^\ell - s_{j-1,k-1}^\ell}{\Delta x} + \frac{s_{jk}^\ell - s_{j,k-1}^\ell}{\Delta x} \right) \quad (6.49)$$

the centered discretizations of $\frac{\partial s}{\partial x}$ and $\frac{\partial s}{\partial y}$ evaluated at point $(j - \frac{1}{2}, k - \frac{1}{2})\Delta x$ and

$$\begin{aligned} (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k-\frac{1}{2}} = & \frac{1}{2} \left(\frac{s_{j,k-1}^\ell - s_{j-1,k-1}^\ell}{\Delta x} \frac{\lambda_{j,k-1}^\ell - \lambda_{j-1,k-1}^\ell}{\Delta x} + \frac{s_{jk}^\ell - s_{j-1,k}^\ell}{\Delta x} \frac{\lambda_{jk}^\ell - \lambda_{j-1,k}^\ell}{\Delta x} \right. \\ & \left. + \frac{s_{j-1,k}^\ell - s_{j-1,k-1}^\ell}{\Delta x} \frac{\lambda_{j-1,k}^\ell - \lambda_{j-1,k-1}^\ell}{\Delta x} + \frac{s_{j,k}^\ell - s_{j,k-1}^\ell}{\Delta x} \frac{\lambda_{jk}^\ell - \lambda_{j,k-1}^\ell}{\Delta x} \right) \end{aligned} \quad (6.50)$$

the centered discretization of $\nabla s \cdot \nabla \lambda$ at the same point. The formulation of the dual problem requires the discretization of three physical quantities, namely $\nabla \cdot (\mathcal{D} \nabla \lambda)$, $((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s) \nabla s \cdot \nabla \lambda$, and $\nabla \cdot (\mathcal{G} \nabla s \cdot \nabla \lambda \nabla s)$. With the notations introduced above, write

$$\left[\nabla_h \cdot (\mathcal{D} \nabla_h \lambda) \right]_{jk}^\ell = \frac{1}{\Delta x} \left[\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{\lambda_{j+1,k}^\ell - \lambda_{jk}^\ell}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{\lambda_{jk}^\ell - \lambda_{j-1,k}^\ell}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{\lambda_{j,k+1}^\ell - \lambda_{jk}^\ell}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{\lambda_{jk}^\ell - \lambda_{j,k-1}^\ell}{\Delta x} \right], \quad (6.51)$$

the discretization at point $(j, k)\Delta x$ of $\nabla \cdot (\mathcal{D} \nabla \lambda)$, where the diffusivities are defined by (3.21)-(3.22),

$$\begin{aligned} \left[((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s) \nabla_h s \cdot \nabla_h \lambda \right]_{jk}^\ell = & \frac{1}{4} \left[\left((n+2)(\mathcal{F}_f)_{j-\frac{1}{2}, k-\frac{1}{2}} + (n+1)(\mathcal{F}_s)_{j-\frac{1}{2}, k-\frac{1}{2}} \right) (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k-\frac{1}{2}} \right. \\ & + \left((n+2)(\mathcal{F}_f)_{j-\frac{1}{2}, k+\frac{1}{2}} + (n+1)(\mathcal{F}_s)_{j-\frac{1}{2}, k+\frac{1}{2}} \right) (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k+\frac{1}{2}} \\ & + \left((n+2)(\mathcal{F}_f)_{j+\frac{1}{2}, k-\frac{1}{2}} + (n+1)(\mathcal{F}_s)_{j+\frac{1}{2}, k-\frac{1}{2}} \right) (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k-\frac{1}{2}} \\ & \left. + \left((n+2)(\mathcal{F}_f)_{j+\frac{1}{2}, k+\frac{1}{2}} + (n+1)(\mathcal{F}_s)_{j+\frac{1}{2}, k+\frac{1}{2}} \right) (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k+\frac{1}{2}} \right], \end{aligned} \quad (6.52)$$

the discretization of $((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s) \nabla s \cdot \nabla \lambda$, and

$$\begin{aligned} \left[\nabla_h \cdot (\mathcal{G} \nabla_h s \cdot \nabla_h \lambda \nabla_h s) \right]_{jk}^\ell = & \frac{1}{2\Delta x} \left[\mathcal{G}_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_x^c s)_{j+\frac{1}{2}, k-\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k-\frac{1}{2}} - \mathcal{G}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_x^c s)_{j-\frac{1}{2}, k-\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k-\frac{1}{2}} \right. \\ & + \mathcal{G}_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_x^c s)_{j+\frac{1}{2}, k+\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k+\frac{1}{2}} - \mathcal{G}_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_x^c s)_{j-\frac{1}{2}, k+\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k+\frac{1}{2}} \\ & + \mathcal{G}_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_y^c s)_{j-\frac{1}{2}, k+\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k+\frac{1}{2}} - \mathcal{G}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_y^c s)_{j-\frac{1}{2}, k-\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k-\frac{1}{2}} \\ & \left. + \mathcal{G}_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_y^c s)_{j+\frac{1}{2}, k+\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k+\frac{1}{2}} - \mathcal{G}_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_y^c s)_{j+\frac{1}{2}, k-\frac{1}{2}} (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k-\frac{1}{2}} \right], \end{aligned} \quad (6.53)$$

the discretization of $\nabla \cdot (\mathcal{G} \nabla s \cdot \nabla \lambda \nabla s)$, where \mathcal{F}_f , \mathcal{F}_s , and \mathcal{G} are defined by (A.30). Moreover, the expression of the gradient uses the discretization of $\mathcal{K}^{n+1, n-1} (\nabla s \cdot \nabla \lambda)$, which is

$$\begin{aligned} \left[\mathcal{K}^{n+1, n-1} (\nabla_h s \cdot \nabla_h \lambda) \right]_{jk}^\ell = & \frac{1}{4} \left[\mathcal{K}_{j-\frac{1}{2}, k-\frac{1}{2}}^{n+1, n-1}(\ell) (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k-\frac{1}{2}} + \mathcal{K}_{j+\frac{1}{2}, k-\frac{1}{2}}^{n+1, n-1}(\ell) (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k-\frac{1}{2}} \right. \\ & \left. + \mathcal{K}_{j-\frac{1}{2}, k+\frac{1}{2}}^{n+1, n-1}(\ell) (\delta s \cdot \delta \lambda)_{j-\frac{1}{2}, k+\frac{1}{2}} + \mathcal{K}_{j+\frac{1}{2}, k+\frac{1}{2}}^{n+1, n-1}(\ell) (\delta s \cdot \delta \lambda)_{j+\frac{1}{2}, k+\frac{1}{2}} \right]. \end{aligned} \quad (6.54)$$

In the semi-implicit time discretization (3.19), the right-hand side of (6.40) reads therefore

$$(\tilde{\mathcal{T}}_1)_{jk}^\ell = \left[\nabla_h \cdot (\mathcal{D} \nabla_h \bar{\lambda}) \right]_{jk}^\ell \quad (6.55)$$

and the right-hand side of (6.41) reads

$$(\tilde{\mathcal{T}}_2)_{jk}^\ell = - \left[\left((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s \right) \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \right]_{jk}^\ell + (n-1) \left[\nabla_h \cdot \left(\mathcal{G} \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \nabla_h s \right) \right]_{jk}^\ell, \quad (6.56)$$

where the discretizations (6.51), (6.52), and (6.53) are applied. Once the state and costate variables have been determined, the gradient of the cost functional

$$J_h(\mathbf{b}) = \mathcal{J}_h(\mathcal{H}(\mathbf{b}), \mathbf{b}) \quad (6.57)$$

can be computed with

$$\begin{aligned} \frac{dJ_h}{db_{jk}} &= \lambda_{jk}^0 + \left((s_f)_{jk} - s^{\text{obs}}(\xi_j, \zeta_k) \right) + \frac{\partial \mathcal{R}}{\partial b_{jk}} - \Delta t + \sum_{\ell=0}^M \sum_{il} \vartheta(\overline{\mathcal{H}}_{il}^\ell) (\nabla_{\mathbf{b}} \overline{\mathcal{B}})_{ij;kl}^\ell \overline{\lambda}_{il}^\ell \\ &\quad - \Delta t \sum_{\ell=0}^M \left[(n-1) \left[\nabla_h \cdot \left(\mathcal{G} \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \nabla_h s \right) \right]_{jk}^\ell + \left[\nabla_h \cdot (\mathcal{D} \nabla_h \lambda) \right]_{jk}^\ell + (\mathcal{C} \rho g)^n \vartheta(z_{\text{sl}} - b)_{jk} \left[\mathcal{K}^{n+1, n-1} \left(\nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \right) \right]_{jk}^\ell \right] \end{aligned} \quad (6.58)$$

where the discretizations (6.51), (6.53), and (6.54) are applied, \mathcal{C} is the sliding constant defined by (2.10), $(\nabla_{\mathbf{b}} \overline{\mathcal{B}})_{ij;kl}^\ell$ is given by (6.47), and

$$\begin{aligned} \frac{d\mathcal{R}_h}{db_{jk}} &= \frac{r_1}{2\Delta x^2} \left(4b_{jk} - b_{j-1,k} - b_{j+1,k} - b_{j,k-1} - b_{j,k+1} \right) \\ &\quad + \frac{r_2}{2\Delta x^4} \left(20b_{jk} - 8(b_{j+1,k} + b_{j-1,k} + b_{j,k+1} + b_{j,k-1}) + 2(b_{j+1,k+1} + b_{j-1,k-1} + b_{j-1,k+1} + b_{j+1,k-1}) \right. \\ &\quad \left. + b_{j+2,k} + b_{j-2,k} + b_{j,k+2} + b_{j,k-2} \right). \end{aligned} \quad (6.59)$$

In the implicit time discretization (3.27), the right-hand side of (6.40) is

$$(\tilde{\mathcal{T}}_1)_{jk}^\ell = \frac{1}{2} \left[\left[\nabla_h \cdot (\mathcal{D} \nabla_h \bar{\lambda}) \right]_{jk}^\ell - \left[\left((n+2)\overline{\mathcal{F}}_f + (n+1)\overline{\mathcal{F}}_s \right) \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \right]_{jk}^\ell + (n-1) \left[\nabla_h \cdot \left(\mathcal{G} \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \nabla_h \bar{s} \right) \right]_{jk}^\ell \right] \quad (6.60)$$

and the right-hand side of (6.41) is the same but with \bar{s} replaced with s everywhere,

$$(\tilde{\mathcal{T}}_2)_{jk}^\ell = \frac{1}{2} \left[\left[\nabla_h \cdot (\mathcal{D} \nabla_h \bar{\lambda}) \right]_{jk}^\ell - \left[\left((n+2)\mathcal{F}_f + (n+1)\mathcal{F}_s \right) \nabla_h s \cdot \nabla_h \bar{\lambda} \right]_{jk}^\ell + (n-1) \left[\nabla_h \cdot \left(\mathcal{G} \nabla_h s \cdot \nabla_h \bar{\lambda} \nabla_h s \right) \right]_{jk}^\ell \right], \quad (6.61)$$

where the discretizations (6.51), (6.52), and (6.53) are applied. Once the state and costate variables have been determined, the gradient of the cost functional can be computed with

$$\begin{aligned} \frac{dJ_h}{db_{jk}} &= \lambda_{jk}^0 + \left((s_f)_{jk} - s^{\text{obs}}(\xi_j, \zeta_k) \right) + \frac{\partial \mathcal{R}_h}{\partial b_{jk}} - \Delta t \sum_{\ell=0}^M \sum_{il} \vartheta(\overline{\mathcal{H}}_{il}^\ell) (\nabla_{\mathbf{b}} \overline{\mathcal{B}})_{ij;kl}^\ell \overline{\lambda}_{il}^\ell \\ &\quad - \Delta t \sum_{\ell=0}^M \left[\frac{n-1}{2} \left[\left[\nabla_h \cdot \left(\mathcal{G} \nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \nabla_h \bar{s} \right) \right]_{jk}^\ell + \left[\nabla_h \cdot \left(\mathcal{G} \nabla_h s \cdot \nabla_h \bar{\lambda} \nabla_h s \right) \right]_{jk}^\ell \right] + \frac{1}{2} \left[\left[\nabla_h \cdot (\mathcal{D} \nabla_h \bar{\lambda}) \right]_{jk}^\ell + \left[\nabla_h \cdot (\mathcal{D} \nabla_h \lambda) \right]_{jk}^\ell \right] \right. \\ &\quad \left. + (\mathcal{C} \rho g)^n \vartheta(z_{\text{sl}} - b)_{jk} \left[\left[\mathcal{K}^{n+1, n-1} \left(\nabla_h s \cdot \nabla_h \bar{\lambda} \right) \right]_{jk}^\ell + \left[\mathcal{K}^{n+1, n-1} \left(\nabla_h \bar{s} \cdot \nabla_h \bar{\lambda} \right) \right]_{jk}^\ell \right] \right], \end{aligned} \quad (6.62)$$

where the discretizations (6.51), (6.53), and (6.54) are applied, \mathcal{C} is the sliding constant defined by (2.10), $(\nabla_{\mathbf{b}} \overline{\mathcal{B}})_{ij;kl}^\ell$ is given by (6.47), and the gradient of the regularization \mathcal{R}_h is given by (6.59).

The gradients correspond to the discretizations of the continuous expression

$$\begin{aligned} \frac{dJ}{db} \hat{b} = & \int_{\Omega_{\perp}} \left[(s_f - s^{\text{obs}} + \lambda) \Big|_{t=t_i} + \frac{\partial \mathcal{R}}{\partial b} \right] \hat{b} d\mathbf{x} - \int_{t_i}^{t_f} \int_{\Omega_{\perp}} \lambda \frac{\partial \mathcal{R}}{\partial b} \hat{b} d\mathbf{x} dt \\ & - \int_{t_i}^{t_f} \int_{\Omega_{\perp}} \nabla \cdot \left[(n-1) \mathcal{G} \nabla s \cdot \nabla \lambda \nabla s + \mathcal{D} \nabla \lambda \right] \hat{b} d\mathbf{x} dt - \int_{t_i}^{t_f} \int_{\Omega_{\perp}} (\rho g \mathcal{C})^n \vartheta(z_{\text{sl}} - b) \mathcal{H}^{n+1} \|\nabla s\|^{n-1} \nabla s \cdot \nabla \lambda \hat{b} d\mathbf{x} dt \end{aligned} \quad (6.63)$$

for any continuous function \hat{b} .

In the chosen time frame $[t_i, t_f]$, it may be that the surface topography is known at several times. This can be taken into account in the computations by updating cost functional (6.33) so that it includes this information. Let $0 \leq \ell_m^s \leq M+1$, $m = 1, 2, \dots, N_s$, be the time steps where the surface topography is known and denoted by $(s^{\text{obs}})^{\ell_m^s}$. Then, cost functional (6.33) is replaced with

$$\mathcal{J}_h(\mathcal{H}, \mathbf{b}) = \frac{1}{2} \sum_{m=1}^{N_s} \sum_{jk} \left(b_{jk} + \mathcal{H}_{jk}^{\ell_m^s} - (s^{\text{obs}})^{\ell_m^s}_{jk} \right)^2 + \mathcal{R}_h(\mathbf{b}). \quad (6.64)$$

This change affects the dual equation (6.41) and the cost functional's gradients (6.58) and (6.62). The dual equation (6.41) changes to

$$-\frac{\bar{\lambda}_{jk}^{\ell} - \lambda_{jk}^{\ell}}{\Delta t} = (\tilde{\mathcal{T}}_2)_{jk}^{\ell} + \frac{1}{\Delta t} \left(s^{\text{obs}}(\xi_j, \zeta_k, t^{\ell}) - b_{jk} - \mathcal{H}_{jk}^{\ell} \right), \quad (6.65)$$

for every time step ℓ where the surface topography is known to be $s^{\text{obs}}|_{t=t^{\ell}}$ and the gradients (6.58) and (6.62) are augmented by the amount

$$\frac{dJ_h}{db_{jk}} + = \sum_{m=1}^{N_s} \sum_{jk} \left(b_{jk} + \mathcal{H}_{jk}^{\ell_m^s} - s^{\text{obs}}(\xi_j, \zeta_k, t^{\ell_m^s}) \right). \quad (6.66)$$

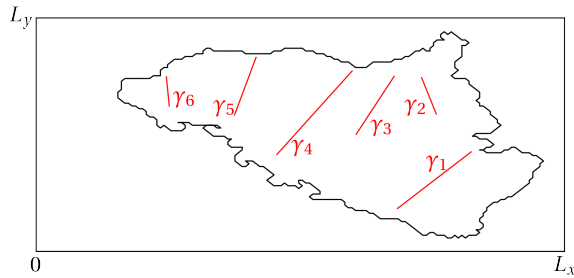


Figure 6.2 – Known bedrock topography profiles γ_i inside the glacier's map domain.

Furthermore, the method can also take into account known bedrock topography data. Imagine bedrock topography profiles γ_i are available inside the glacier's map domain, as illustrated by the example Figure 6.2. Let I_b be the set of indices (j, k) in the map domain $\tilde{\Omega}_{\perp}$ where an observation b^{obs} of the bedrock topography is given. Then, the cost functional (6.33) is augmented with the contribution

$$\mathcal{J}_h(\mathcal{H}, \mathbf{b}) + = \frac{1}{2} \sum_{j,k \in I_b} \left(b_{jk} - b^{\text{obs}}(\xi_j, \zeta_k) \right)^2, \quad (6.67)$$

and its gradient with

$$\frac{dJ_h}{db_{jk}} = \begin{cases} b_{jk} - b^{\text{obs}}(\xi_j, \zeta_k), & \text{if } (j, k) \in I_b \\ 0, & \text{otherwise,} \end{cases} \quad (6.68)$$

in both the semi-implicit and implicit time discretizations.

As in the two-dimensional case, the minimization of J_h is performed by finding the control \mathbf{b} where the gradient of J_h vanishes in the computational domain with a quasi-Newton method. The Hessian matrix of J_h is approximated using the BFGS algorithm [15, 35, 47, 107]. All the relevant implementation details are given in Appendix B.1. The method is summarized in Algorithms 7 and 8.

Algorithm 7 Shape optimization algorithm (semi-implicit time discretization)

set glacier geometry with initial bedrock b^0 and surface s_i
while $|J_h(\mathbf{b}^{m+1}) - J_h(\mathbf{b}^m)| > \delta$ **do**
 solve the primal problem (3.14), (3.15), (3.16), and (3.18) with (3.19)
 solve the dual problem (6.38), (6.39), (6.40), and (6.41) with (6.55) and (6.56)
 compute the gradient (6.58)
 compute

$$\mathbf{b}^{m+1} = \mathbf{b}^m + \left(\frac{d^2 J_h}{d\mathbf{b}^2}(\mathbf{b}^m) \right)^{-1} \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}^m),$$

where the Hessian is approximated with the BFGS method

project the vector \mathbf{b}^{m+1} onto the space of admissible controls $\mathcal{U}_h^{\text{ad}}$.

end while

Algorithm 8 Shape optimization algorithm (implicit time discretization)

set glacier geometry with initial bedrock b^0 and surface s_i
while $|J_h(\mathbf{b}^{m+1}) - J_h(\mathbf{b}^m)| > \delta$ **do**
 solve the primal problem (3.14), (3.15), (3.16), and (3.18) with (3.27)
 solve the dual problem (6.38), (6.39), (6.40), and (6.41) with (6.60) and (6.61)
 compute the gradient (6.62)
 compute

$$\mathbf{b}^{m+1} = \mathbf{b}^m + \left(\frac{d^2 J_h}{d\mathbf{b}^2}(\mathbf{b}^m) \right)^{-1} \frac{dJ_h}{d\mathbf{b}}(\mathbf{b}^m),$$

where the Hessian is approximated with the BFGS method

project the vector \mathbf{b}^{m+1} onto the space of admissible controls $\mathcal{U}_h^{\text{ad}}$.

end while

At this point, the method aims at the determination of the glacier's subglacial topography, as the quasi-stationary and inverse methods. However, one may wonder if the surface topography data used in the process may not be sufficient to compute other model features, such as rheology, sliding, and surface mass-balance parameters. This is the subject of the next subsection. Henceforth, only complements to the equations derived above are given, *i.e.* they are not recalled any more but additional building blocks are given instead, for the sake of readability.

6.2.2 Computation of model parameters from surface topography data

The Algorithms 7 and 8 solve the problem of computing the glacier's subglacial topography when the surface topography and mass-balance are known. Now, in practical applications, the model parameters are not well-known, as explained previously in Chapter 4. Therefore, it is natural to address the calculation of these parameters, in the

same time the bedrock topography is inferred. The computation of the model parameters on stationary geometries was successfully attempted for Stokes flows from surface velocity observations [81, 94]. Here, focus is put on transient shallow ice flows.

In the above two and three-dimensional derivations of the cost functional's gradient, the state variable is seen as a function of the bedrock topography only. Consequently, the diffusivities, the surface mass-balance, and the costate variables are also considered as functions of the bedrock topography only. Thereafter, they are functions of the latter topography and all the model parameters. The latter parameters, namely the rate factor A (see expression (2.16)), the melting rates m_1 and m_2 , the equilibrium line altitude z_{ELA} (see expression (2.34)), the sliding altitude z_{sl} , and the sliding constant \mathcal{C} (see expression (2.10)) together with the components of the bedrock topography \mathbf{b} build up the vector \mathbf{q} . From now on, this vector is the “control variable” of the minimization problem. The space of admissible solutions (6.35) is completed to take into account reasonable bounds on the additional control variables. Aim is now to compute the gradient of the cost functional

$$J_h(\mathbf{q}) = \mathcal{J}_h(\mathcal{H}(\mathbf{q}), \mathbf{q}) \quad (6.69)$$

with respect to all the components of \mathbf{q} .

First, let us deal with rheology. The value of Glen's flow exponent is trusted in this thesis to be $n = 3$. The only remaining parameter is the rate factor A . In the semi-implicit time discretization, the derivative of (6.69) with respect to A reads

$$\frac{dJ_h}{dA} = -2 \frac{(\rho g)^n}{n+2} \Delta t \sum_{\ell=0}^M \sum_{jk} \left[\nabla_h \left(\mathcal{K}^{n+2, n-1} \nabla_h \bar{s} \right) \right]_{jk}^{\ell} \bar{\lambda}_{jk}^{\ell}, \quad (6.70)$$

where the term in brackets is the discretization of $\nabla \cdot \left(\mathcal{K}^{n+2, n-1} \nabla \bar{s} \right)$ with formula (6.51) where \mathcal{D} is replaced with \mathcal{K} and s with \bar{s} . In the implicit time discretization, the derivative of (6.69) with respect to A reads

$$\frac{dJ_h}{dA} = -\frac{(\rho g)^n}{n+2} \Delta t \sum_{\ell=0}^M \sum_{jk} \left(\left[\nabla_h \left(\mathcal{K}^{n+2, n-1} \nabla_h s \right) \right]_{jk}^{\ell} + \left[\nabla_h \left(\overline{\mathcal{K}}^{n+2, n-1} \nabla_h \bar{s} \right) \right]_{jk}^{\ell} \right) \bar{\lambda}_{jk}^{\ell}, \quad (6.71)$$

with the same discretization as before for the bracketted terms. Expressions (6.70) and (6.71) are the discretizations of

$$\frac{dJ}{dA} = -2 \frac{(\rho g)^n}{n+2} \int_{t_i}^{t_f} \int_{\Omega_{\perp}} \nabla \cdot \left(\mathcal{K}^{n+2} \|\nabla s\|^{n-1} \nabla s \right) \lambda d\mathbf{x} dt. \quad (6.72)$$

Second, the gradient of J_h with respect to the surface mass-balance parameters is given in both time discretizations by the components

$$\frac{dJ_h}{dz_{\text{ELA}}} = -\Delta t \sum_{\ell=0}^M \sum_{jk} \vartheta(\overline{\mathcal{K}}_{jk}^{\ell}) \left(\frac{m_1 + m_2}{2} - \left(\frac{m_1 - m_2}{2} \right)^2 \frac{b_{jk} + \overline{\mathcal{K}}_{jk}^{\ell} - z_{\text{ELA}}}{\sqrt{\left(\frac{m_1 - m_2}{2} (b_{jk} + \overline{\mathcal{K}}_{jk}^{\ell} - z_{\text{ELA}}) \right)^2 + \varepsilon}} \right) \bar{\lambda}_{jk}^{\ell} \quad (6.73)$$

$$\frac{dJ_h}{dm_1} = -\Delta t \sum_{\ell=0}^M \sum_{jk} \vartheta(\mathcal{K}_{jk}^{\ell}) \left(\frac{b_{jk} + \overline{\mathcal{K}}_{jk}^{\ell} - z_{\text{ELA}}}{2} - \frac{m_1 - m_2}{4} \frac{(b_{jk} + \overline{\mathcal{K}}_{jk}^{\ell} - z_{\text{ELA}})^2}{\sqrt{\left(\frac{m_1 - m_2}{2} (b_{jk} + \overline{\mathcal{K}}_{jk}^{\ell} - z_{\text{ELA}}) \right)^2 + \varepsilon}} \right) \bar{\lambda}_{jk}^{\ell} \quad (6.74)$$

$$\frac{dJ_h}{dm_2} = -\Delta t \sum_{\ell=0}^M \sum_{jk} \vartheta(\mathcal{H}_{jk}^\ell) \left(\frac{b_{jk} + \overline{\overline{\mathcal{H}}}_{jk}^\ell - z_{\text{ELA}}}{2} + \frac{m_1 - m_2}{4} \frac{(b_{jk} + \overline{\overline{\mathcal{H}}}_{jk}^\ell - z_{\text{ELA}})^2}{\sqrt{\left(\frac{m_1 - m_2}{2} (b_{jk} + \overline{\overline{\mathcal{H}}}_{jk}^\ell - z_{\text{ELA}})\right)^2 + \varepsilon}} \right) \overline{\overline{\lambda}}_{jk}^\ell. \quad (6.75)$$

These expressions are the direct derivatives of the surface mass-balance (2.34). Finally, the gradient of J_h with respect to the sliding law parameters is given by the components

$$\frac{dJ_h}{dz_{\text{sl}}} = -(\rho g \mathcal{C})^n \Delta t \sum_{\ell=0}^M \sum_{jk} \left[\nabla_h \cdot \left(\vartheta(z_{\text{sl}} - b) \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell \overline{\overline{\lambda}}_{jk}^\ell \quad (6.76)$$

$$\frac{dJ_h}{d\mathcal{C}} = -\frac{n}{2\mathcal{C}} \Delta t \sum_{\ell=0}^M \sum_{jk} \left[\nabla_h \cdot \left(\Gamma_s \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell \overline{\overline{\lambda}}_{jk}^\ell, \quad (6.77)$$

in the semi-implicit time discretization, where the bracketted terms are the discretizations of $\nabla_h \cdot \left(\vartheta(z_{\text{sl}} - b) \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right)$ and $\nabla_h \cdot \left(\Gamma_s \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right)$ with formula (6.51) adapted appropriately. In the implicit time discretization, this becomes

$$\frac{dJ_h}{dz_{\text{sl}}} = -\frac{(\rho g \mathcal{C})^n}{2} \Delta t \sum_{\ell=0}^M \sum_{jk} \left(\left[\nabla_h \cdot \left(\vartheta(z_{\text{sl}} - b) \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell + \left[\nabla_h \cdot \left(\vartheta(z_{\text{sl}} - b) \overline{\overline{\mathcal{K}}}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell \right) \overline{\overline{\lambda}}_{jk}^\ell \quad (6.78)$$

$$\frac{dJ_h}{d\mathcal{C}} = -\frac{n}{2\mathcal{C}} \Delta t \sum_{\ell=0}^M \sum_{jk} \left(\left[\nabla_h \cdot \left(\Gamma_s \mathcal{K}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell + \left[\nabla_h \cdot \left(\Gamma_s \overline{\overline{\mathcal{K}}}^{n+1, n-1} \nabla_h \bar{s} \right) \right]_{jk}^\ell \right) \overline{\overline{\lambda}}_{jk}^\ell, \quad (6.79)$$

with the same discretizations as before for the bracketted terms. These derivatives correspond to the discretizations of

$$\frac{dJ}{dz_{\text{sl}}} = -(\mathcal{C} \rho g)^n \int_{t_i}^{t_f} \int_{\Omega_\perp} \nabla \cdot \left(\vartheta(z_{\text{sl}} - b) \mathcal{K}^{n+1} \|\nabla s\|^{n-1} \nabla s \right) \lambda \, d\mathbf{x} dt \quad (6.80)$$

$$\frac{dJ}{d\mathcal{C}} = -\frac{n}{\mathcal{C}} \int_{t_i}^{t_f} \int_{\Omega_\perp} \nabla \cdot \left(\mathcal{D}_s \nabla s \right) \lambda \, d\mathbf{x} dt. \quad (6.81)$$

6.2.3 Incorporation of velocity data

Contrary to the previously described quasi-stationary and transient inverse methods, the shape optimization algorithm can include surface velocity data. Let $0 \leq \ell_m^u \leq M+1$, $m = 1, 2, \dots, N_u$, be the time steps where the surface velocity is known and denoted by $u^{\text{obs}} \Big|_{t=t^{\ell_m^u}}$. Let $I_u^{\ell_m^u}$ be the set of indices (j, k) in the map domain $\tilde{\Omega}_\perp$ corresponding to the locations where an observation u^{obs} of the surface velocity is given at time step ℓ_m^u . Then, the cost functional (6.33) is augmented by the dependence on \mathcal{U} and the term

$$\mathcal{J}_h(\mathcal{H}, \mathcal{U}, \mathbf{b}) + = \frac{1}{2} (t_f - t_i)^2 \sum_{m=1}^{N_u} \sum_{j,k \in I_u^{\ell_m^u}} \left(\mathcal{U}_{jk}^{\ell_m^u} - u^{\text{obs}}(\xi_j, \zeta_k, t^{\ell_m^u}) \right)^2, \quad (6.82)$$

where \mathcal{U}_{jk}^ℓ is the state variable (3.31) that represents the model's surface velocity. Let \mathbf{v}^ℓ be the costate variable of the surface velocity \mathcal{U}^ℓ . Then, the Lagrangian (6.36) is augmented with the dependence on \mathcal{U} and \mathbf{v} and the term

$$\mathcal{L}_h + = \sum_{m=1}^{N_u} \left[\mathcal{U}^{\ell_m^u} - \mathbf{u}^{\ell_m^u} \right]^T \mathbf{v}^{\ell_m^u}. \quad (6.83)$$

Chapter 6. Optimal control of the shallow ice model

Note that this last scalar product is not necessary and could be actually merged within \mathcal{J}_h . It is kept here for the sake of extensibility to higher order models. Such a modification of the Lagrangian implies that almost all the previously computed gradients must be updated, as well as the dual equation (6.41).

First, a new dual equation arises, namely

$$\mathbf{v}_{jk}^\ell = (t_f - t_i)^2 \begin{cases} u^{\text{obs}}(\xi_j, \zeta_k, t^\ell) - \mathcal{U}_{jk}^\ell, & \text{if } (j, k) \in I_u^\ell \\ 0, & \text{otherwise,} \end{cases} \quad (6.84)$$

that gives a value for the costate variable of the surface velocity, for all time iterations ℓ where a surface velocity observation is available. Second, the final condition (6.38) of the dual problem is updated to

$$\lambda_{jk}^{M+1} = [\mathcal{P}v]_{jk}^{M+1} - \left[\nabla_h \cdot (\mathcal{Q}v \nabla_h s) \right]_{jk}^{M+1}, \quad (6.85)$$

where the first term is the discretization of $\mathcal{P}v$ at time step $\ell = M + 1$, *i.e.*

$$[\mathcal{P}v]_{jk}^\ell = \frac{1}{4} \left[\mathcal{P}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell v_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell + \mathcal{P}_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell v_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell + \mathcal{P}_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell v_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell + \mathcal{P}_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell v_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell \right], \quad (6.86)$$

where

$$\mathcal{P}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell = \left((n+2)\Gamma \mathcal{H}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell + n(\Gamma_s)_{j-\frac{1}{2}, k-\frac{1}{2}} \right) \mathcal{K}_{j-\frac{1}{2}, k-\frac{1}{2}}^{n-1, n}(\ell) \quad (6.87)$$

and

$$v_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell = \frac{1}{4} \left(v_{j-1, k-1}^\ell + v_{j, k-1}^\ell + v_{j-1, k}^\ell + v_{j, k}^\ell \right). \quad (6.88)$$

The second term of (6.85) is the discretization of $\nabla \cdot (\mathcal{Q}v \nabla s)$ at time step $\ell = M + 1$

$$\begin{aligned} \left[\nabla_h \cdot (\mathcal{Q}v \nabla_h s) \right]_{jk}^\ell &= \frac{n-1}{2\Delta x} \left[\mathcal{Q}_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_x^c s)_{j+\frac{1}{2}, k-\frac{1}{2}} v_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell - \mathcal{Q}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_x^c s)_{j-\frac{1}{2}, k-\frac{1}{2}} v_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell \right. \\ &\quad + \mathcal{Q}_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_x^c s)_{j+\frac{1}{2}, k+\frac{1}{2}} v_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell - \mathcal{Q}_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_x^c s)_{j-\frac{1}{2}, k+\frac{1}{2}} v_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell \\ &\quad + \mathcal{Q}_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_y^c s)_{j-\frac{1}{2}, k+\frac{1}{2}} v_{j-\frac{1}{2}, k+\frac{1}{2}}^\ell - \mathcal{Q}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_y^c s)_{j-\frac{1}{2}, k-\frac{1}{2}} v_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell \\ &\quad \left. + \mathcal{Q}_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell (\delta_y^c s)_{j+\frac{1}{2}, k+\frac{1}{2}} v_{j+\frac{1}{2}, k+\frac{1}{2}}^\ell - \mathcal{Q}_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell (\delta_y^c s)_{j+\frac{1}{2}, k-\frac{1}{2}} v_{j+\frac{1}{2}, k-\frac{1}{2}}^\ell \right], \end{aligned} \quad (6.89)$$

where $\delta_x^c s$ and $\delta_y^c s$ are the discretizations (6.48) and (6.49) respectively, and

$$\mathcal{Q}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell = n \left(\frac{n+2}{n+1} \Gamma \mathcal{H}_{j-\frac{1}{2}, k-\frac{1}{2}}^\ell + (\Gamma_s)_{j-\frac{1}{2}, k-\frac{1}{2}} \right) \mathcal{K}_{j-\frac{1}{2}, k-\frac{1}{2}}^{n, n-2}(\ell). \quad (6.90)$$

The dual equations (6.41) and (6.65) are replaced with

$$-\frac{\bar{\lambda}_{jk}^\ell - \lambda_{jk}^\ell}{\Delta t} = (\tilde{\mathcal{J}}_2)_{jk}^\ell + \frac{1}{\Delta t} \left(s^{\text{obs}}(\xi_j, \zeta_k, t^\ell) - b_{jk} - \mathcal{H}_{jk}^\ell \right) + [\mathcal{P}v]_{jk}^\ell - \left[\nabla_h \cdot (\mathcal{Q}v \nabla_h s) \right]_{jk}^\ell \quad (6.91)$$

for every time step ℓ where surface velocity data is available. Third, the gradient of the cost functional with respect

to \mathbf{b} must be corrected by the amount

$$\frac{dJ_h}{db_{jk}} = \sum_{m=0}^{N_u} \left((\mathcal{C}\rho g)^n \vartheta(z_{sl} - b)_{jk} [\mathcal{K}^{n,n} v]_{jk}^{\ell_m^u} + \left[\nabla_h \cdot (\mathcal{Q} v \nabla_h s) \right]_{jk}^{\ell_m^u} \right), \quad (6.92)$$

where the first term uses the discretization

$$\begin{aligned} [\mathcal{K}^{n,n} v]_{jk}^{\ell} = \frac{1}{4} & \left[\mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell) v_{j-\frac{1}{2},k-\frac{1}{2}}^{\ell} + \mathcal{K}_{j+\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell) v_{j+\frac{1}{2},k-\frac{1}{2}}^{\ell} \right. \\ & \left. + \mathcal{K}_{j-\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell) v_{j-\frac{1}{2},k+\frac{1}{2}}^{\ell} + \mathcal{K}_{j+\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell) v_{j+\frac{1}{2},k+\frac{1}{2}}^{\ell} \right]. \end{aligned} \quad (6.93)$$

of $\mathcal{K}^{n,n} v$ and the second term the discretization (6.89). Fourth, the derivative of J_h with respect to the rate factor must be augmented by the amount

$$\frac{dJ_h}{dA} = \frac{1}{2} \frac{(\rho g)^n}{n+1} \sum_{m=0}^{N_u} \sum_{jk} \left(\mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n+1,n}(\ell_m^u) + \mathcal{K}_{j+\frac{1}{2},k-\frac{1}{2}}^{n+1,n}(\ell_m^u) + \mathcal{K}_{j-\frac{1}{2},k+\frac{1}{2}}^{n+1,n}(\ell_m^u) + \mathcal{K}_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1,n}(\ell_m^u) \right) v_{jk}^{\ell_m^u}. \quad (6.94)$$

Fifth, the derivative of J_h with respect to the sliding altitude must be updated with

$$\begin{aligned} \frac{dJ_h}{dz_{sl}} = \frac{1}{4} \sum_{m=0}^{N_u} \sum_{jk} & \left(\vartheta(z_{sl} - b)_{j-\frac{1}{2},k-\frac{1}{2}} \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell_m^u) + \vartheta(z_{sl} - b)_{j+\frac{1}{2},k-\frac{1}{2}} \mathcal{K}_{j+\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell_m^u) \right. \\ & \left. + \vartheta(z_{sl} - b)_{j-\frac{1}{2},k+\frac{1}{2}} \mathcal{K}_{j-\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell_m^u) + \vartheta(z_{sl} - b)_{j+\frac{1}{2},k+\frac{1}{2}} \mathcal{K}_{j+\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell_m^u) \right) v_{jk}^{\ell_m^u}. \end{aligned} \quad (6.95)$$

Finally, the derivative of J_h with respect to the sliding constant is changed with

$$\begin{aligned} \frac{dJ_h}{d\mathcal{C}} = \frac{1}{4} \sum_{m=0}^{N_u} \sum_{jk} & \left((\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell_m^u) + (\Gamma_s)_{j+\frac{1}{2},k-\frac{1}{2}} \mathcal{K}_{j+\frac{1}{2},k-\frac{1}{2}}^{n,n}(\ell_m^u) \right. \\ & \left. + (\Gamma_s)_{j-\frac{1}{2},k+\frac{1}{2}} \mathcal{K}_{j-\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell_m^u) + (\Gamma_s)_{j+\frac{1}{2},k+\frac{1}{2}} \mathcal{K}_{j+\frac{1}{2},k+\frac{1}{2}}^{n,n}(\ell_m^u) \right) v_{jk}^{\ell_m^u}. \end{aligned} \quad (6.96)$$

6.2.4 Automatic differentiation

The primal–dual method supplies a backward sensitivity of the cost functional with respect to all the parameters. Another, more direct way to obtain the cost functional's gradient is through its forward sensitivity. It consists in applying directly the chain rule to the derivative of the cost functional with respect to all the control parameters, which is performed by automatic differentiation. Much less hand coding is required in this case and the resulting implementation is extensible to take into account various surface mass-balance models and sliding laws. For the sake of simplicity, let us focus on the computation of the bedrock topography from the observation of the surface topography at final time. Then, the cost functional to be minimized is defined by (6.33) and (6.57) and the purpose is to compute its derivatives with respect to all the bedrock topography elevations b_{jk} . By the chain rule,

$$\frac{dJ_h}{db_{ij}} = b_{ij} + \mathcal{H}_{ij}^{M+1} - s^{\text{obs}}(\xi_i, \zeta_j) + \sum_{kl} \left(b_{kl} + \mathcal{H}_{kl}^{M+1} - s^{\text{obs}}(\xi_k, \zeta_l) \right) \frac{d\mathcal{H}_{kl}^{M+1}}{db_{ij}}, \quad (6.97)$$

where $\frac{d\mathcal{H}_{kl}^{M+1}}{db_{ij}}$ is the output of the following scheme: Initialize (3.14) and its derivative

$$\frac{d\mathcal{H}_{kl}^0}{db_{ij}} = -\delta_{ik} \delta_{jl}. \quad (6.98)$$

Then, first solve the advection-diffusion equation (3.15) and its derivative

$$\frac{\frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}} - \frac{d\mathcal{H}_{kl}^\ell}{db_{ij}}}{\Delta t} = \frac{\partial \mathcal{T}_{kl}^\ell}{\partial b_{ij}^\ell} + \sum_{r,s} \left(\frac{\partial \mathcal{T}_{kl}^\ell}{\partial \mathcal{H}_{rs}^\ell} \frac{d\mathcal{H}_{rs}^\ell}{db_{ij}} + \frac{\partial \mathcal{T}_{kl}^\ell}{\partial \overline{\mathcal{H}}_{rs}^\ell} \frac{d\overline{\mathcal{H}}_{rs}^\ell}{db_{ij}} \right) \quad (6.99)$$

for the coefficients $\frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}}$; second, solve the mass-balance equation (3.16) and its derivative

$$\frac{\frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}} - \frac{d\mathcal{H}_{kl}^\ell}{db_{ij}}}{\Delta t} = \delta(\overline{\mathcal{H}}_{kl}^\ell) \frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}} \overline{\mathcal{B}}_{kl}^\ell + \vartheta(\overline{\mathcal{H}}_{kl}^\ell) \left(\sum_{r,s} \frac{\partial \overline{\mathcal{B}}_{kl}^\ell}{\partial \overline{\mathcal{H}}_{rs}^\ell} \frac{d\overline{\mathcal{H}}_{rs}^\ell}{db_{ij}} + \frac{\partial \overline{\mathcal{B}}_{kl}^\ell}{\partial b_{ij}} \right) \quad (6.100)$$

for the coefficients $\frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}}$; finally, solve the projection equation (3.18) and its derivative

$$\frac{d\mathcal{H}_{kl}^{\ell+1}}{db_{ij}} = \vartheta(\overline{\mathcal{H}}_{kl}^\ell) \frac{d\overline{\mathcal{H}}_{kl}^\ell}{db_{ij}} \quad (6.101)$$

for the coefficients $\frac{d\mathcal{H}_{kl}^{\ell+1}}{db_{ij}}$. Equation (6.99) is solved by automatically differentiating the computer scientific routines that build the linear system (3.15). Equation (6.100) is solved by automatically differentiating the surface mass-balance equation (3.16) and equation (6.101) by automatically differentiating the projection equation (3.18). At each time step, the solution to the forward scheme is first computed, as it is used in the differentiated equations. Then, the latter equations are solved.

As long as the number of parameters b_{jk} is smaller than the number of time steps, the automatic differentiation algorithm (ADA) is less computer memory-demanding than the previously developed primal–dual method. Indeed, the primal – dual method computing the backward sensitivity, it must store the data calculated at all time steps and then use them during the calculation of the cost functional's gradient. The ADA deals with the forward sensitivity and hence does not need to store the data arising at each time step, but instead must store the sensitivity of every grid point with respect to every grid point for a given time step. Once the linear systems for the current time step are solved, the computed sensitivities can be replaced with the new ones needed by the next time step.

Furthermore, many libraries exist that allow to automatically differentiate an existing code implementing a forward model (e.g. Sacado, from the Trilinos project [59]). Therefore, the code's differentiation only consists in the application of existing, optimized code templates that considerably reduce the amount of coding. For instance, whatever change is operated in the surface mass-balance or the sliding model, no modification of the code is necessary, as the new models are differentiated automatically, which makes such a procedure versatile and flexible.

However, with the forward ADA, the sensitivity of the cost functional to each of the other parameters must be computed for every time step for each parameter. In other words, for each grid point, a large linear system has to be solved, making this method very slow when the number of parameters is large, as is the case in shape optimization. For this reason, this method is not developed further in this thesis. A reverse mode, automatic differentiation algorithm may be used in this case [57, 46, 100], but this is too sophisticated for an application to this simple ice flow model.

6.3 Numerical results

In this section, various numerical experiments are performed that test the two and three-dimensional algorithms presented in the previous sections. The purpose is to validate the developed numerical schemes on synthetical data. Their sensitivity to the regularization parameters, the surface topography and mass-balance, and model parameters is presented in Section 6.4. The application of the methods to real-world data is done in Section 6.5.

Hereafter, the two-dimensional Algorithm 6 is first validated on selected profiles of the geometries of Flask, Muragl, Silvretta, and Rhône glaciers in the case of stationary and transient evolutions. The method's performance is compared to that of the stationary and transient inverse methods of Sections 5.2 and 5.4. Then, the three-dimensional shape optimization algorithm is validated on synthetical, transient geometries inspired from Silvretta and Gries glaciers. The results of the validation are compared to their counterparts in the quasi-stationary and transient inverse methods of Sections 5.5.2 and 5.5.3. Then, various numerical experiments are presented where the bedrock topography is reconstructed along with part of the model parameters from surface topography and velocity data.

6.3.1 Two-dimensional formulation

It is here addressed how convergence rate, number of iterations, computational time, and accuracy are improved with the two-dimensional shape optimization algorithm compared to the stationary and transient inverse methods. The steady algorithm is first compared to the SIM and TIM. No sliding is considered and the surface mass-balance is assumed time-independent. Then, the differences with the TIM in both the fixed and moving-boundary situations are examined. The geometries of the Flask outlet glacier in Antarctica as well as the Muragl, Silvretta, and Rhône mountain glaciers in the Swiss Alps are again the base of the inspiration of the data inverted below. These glaciers are chosen because their subglacial geometry either could be measured [34, 33] or is visible as ice has completely vanished [101]. The geometry of Flask glacier is adapted to produce an additional test case, so that the ice thickness is zero at its boundaries and sliding is neglected. A confrontation with the results obtained with SIM and TIM is carried out. Note that the TIM can be viewed as a special case of the SOA, where the costate variable is zero.

For the sake of numerical validation, available data on two-dimensional glacier evolution according to equation (2.32) are used. In other words, this means that a bedrock topography b^* , an initial surface topography s_i , and a surface mass-balance \mathcal{B} are first chosen, from which the final surface geometry s^{obs} is computed with scheme (3.35). Topography b^* is termed “solution bedrock.” Then, we perturb b^* and give it as an initial guess b^0 to Algorithm 6, whose aim is to recover the solution bedrock b^* . In the Tables below, the computed bedrock topography is denoted by b and the final surface topography obtained with b by s_f . Every computation is done with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, $\delta = 10^{-6}$ m, $\rho = 900$ kg · m⁻³, and $g = 9.81$ m · s⁻². Henceforth, the rheology parameters are set to $n = 3$ and $A = 0.1$ bar⁻³a⁻¹ and the norm of the maximum

$$\|\cdot\| = \max_{x_1 < x < x_2} |\cdot| \quad (6.102)$$

is considered.

Steady geometries

The stationary control problem is solved with the SOA and the results are compared to those obtained with the SIM and TIM. The purpose is to investigate the differences in accuracy and computational time. Table 6.1 gives the errors on the geometries displayed by Figure 6.15 for $r_1 = 1$ and several time frames. Note that for fixed r_1 , the larger the time frame, the more accurate the estimation. Not shown here, the computational time increases with the size of the time frame, because the primal and dual problems require more time steps. On the other hand, if the time

Chapter 6. Optimal control of the shallow ice model

frame is too small, the algorithm has not enough information to deduce the subglacial topography, which explains the large errors in Table 6.1's first column. Although this error is large, the error on the final surface is small due to the problem's ill-posedness. Over this small time frame, even a very wrong bedrock can lead to a final surface close to the desired steady surface geometry.

Table 6.1 – Discrepancy $\|b - b^*\|$ in the bedrock and discrepancy $\|s_f - s^{\text{obs}}\|$ in the final surface topography as a function of the time frame on the four steady geometries represented in Figure 6.15 when the SOA with $r_1 = 1$ is applied. With the size of the time frame, the estimation improves most of the time. A hyphen is printed where the algorithm did not converge.

	$t_f - t_i = 10^{-1} \text{ a}$		$t_f - t_i = 1 \text{ a}$		$t_f - t_i = 10 \text{ a}$	
	$\ b - b^*\ \text{ [m]}$	$\ s_f - s^{\text{obs}}\ \text{ [m]}$	$\ b - b^*\ \text{ [m]}$	$\ s_f - s^{\text{obs}}\ \text{ [m]}$	$\ b - b^*\ \text{ [m]}$	$\ s_f - s^{\text{obs}}\ \text{ [m]}$
Fig. 6.16(a)	8.207e+02	6.325e-02	8.215e+00	2.043e-02	3.353e+00	1.546e-02
Fig. 6.16(b)	6.907e+01	5.410e-02	3.778e+01	5.010e-01	1.554e+00	7.354e-02
Fig. 6.16(c)	8.837e+01	6.176e-02	3.249e+00	1.610e-01	-	-
Fig. 6.16(d)	2.398e+02	1.784e-01	7.526e+00	2.739e-01	4.630e+00	1.118e-01

Table 6.2 – Discrepancies $\|b - b^*\|$ and $\|s_f - s^{\text{obs}}\|$, number of iterations, and computational time needed to converge in both the SIM and SOA. In the SOA, the time frame was set to 1 a. Regularization parameters used (ε corresponds to SIM and r_1 to SOA): (a) $\varepsilon = 10^{-6}$, $r_1 = 8.697 \cdot 10^{-2}$ (b) $\varepsilon = 10^{-4}$, $r_1 = 1.53 \cdot 10^{-5}$ (c) $\varepsilon = 10^{-4}$, $r_1 = 8.751 \cdot 10^{-3}$ (d) $\varepsilon = 10^{-3}$, $r_1 = 2.2258$. Parameter r_1 was chosen in such a way that the subglacial discrepancy $\|b - b^*\|$ is the same in both cases in order to allow for a comparison. On geometry (d), this error is different, because no parameter r_1 could be found to make both discrepancies match exactly. However, this does not affect the comparison significantly. Contrarily to the errors and computational times, the number of iterations of the two methods cannot be compared, since they do not correspond to the same kind of simulation. The solution provided by the SOA is of higher quality, because for the same subglacial discrepancy $\|b - b^*\|$ a smaller surface discrepancy $\|s_f - s^{\text{obs}}\|$ is obtained. Computational times in the SOA are also smaller than in the SIM.

	$\ b - b^*\ \text{ [m]}$		$\ s_f - s^{\text{obs}}\ \text{ [m]}$		n_{its}		$T_{\text{CPU}} \text{ [s]}$	
	SIM	SOA	SIM	SOA	SIM	SOA	SIM	SOA
Fig. 6.16(a)	1.180e+00	1.180e+00	1.090e-02	2.472e-03	1527130	756	3.580e+03	1.014e+03
Fig. 6.16(b)	5.973e-02	5.491e-02	5.297e-02	2.849e-05	96089700	2119	4.383e+04	3.864e+02
Fig. 6.16(c)	1.305e-01	1.305e-01	7.719e-02	2.079e-03	9613360	198	3.093e+03	3.400e+01
Fig. 6.16(d)	2.101e+01	1.374e+01	9.272e-01	3.810e-01	13132500	546	9.974e+03	2.541e+02

With smaller values of r_1 , the SOA can be as accurate as the SIM. This is illustrated by Table 6.2, which shows the discrepancies, number of iterations, and computational times with r_1 chosen such that the same errors in the bedrock topography as with the SIM are obtained for each geometry. Note that columns 5 and 6 of Table 6.2, listing the number of iterations in each case, cannot be compared, because one iteration of the SIM corresponds to one single time step of the glacier evolution towards the steady state, while it is one whole simulation of the glacier evolution over the chosen time frame in the SOA. However, the computational time and the error on the final surface can be confronted. Remark that the computational time needed to get an estimation with the same error on b is everytime smaller when the SOA is used. Moreover, the computed final surface s_f is each time closer to the exact steady surface s^{obs} . Hence, the SOA yields a bedrock topography of a higher quality than the SIM.

Table 6.3 – Discrepancy $\|s_f - s^{\text{obs}}\|$ in the final surface topography, number of iterations, and computational time needed to converge in both the TIM and SOA applied to a steady geometry, when the time frame is 1 a (the discrepancies $\|b - b^*\|$ in the bedrock topography are not listed here but match exactly in both methods). In the TIM, the regularization and fixed-point parameters are set to $\varepsilon = 100$ and $\beta = 0.1$ respectively. As in the comparison with the SIM (see Table 6.2), the quality of the estimation is improved with the SOA. The number of iterations of the two methods can be compared and is much smaller with the SOA. Since fewer evolutions are simulated with SOA, its computational time is smaller. The listed regularization parameters r_1 were chosen such that the discrepancy $\|b - b^*\|$ is the same in both cases.

	r_1	$\ s_f - s^{\text{obs}}\ \text{ [m]}$		n_{its}		$T_{\text{CPU}} \text{ [s]}$	
		TIM	SOA	TIM	SOA	TIM	SOA
Fig. 6.16(a)	$2.480e-01$	$5.906e-01$	$6.333e-03$	17473	854	$1.893e+04$	$1.178e+03$
Fig. 6.16(b)	$1.090e-01$	$5.080e-01$	$3.905e-02$	745	233	$2.190e+02$	$9.110e+01$
Fig. 6.16(c)	$3.270e+00$	$1.575e+01$	$5.001e-01$	2980	160	$6.315e+02$	$3.449e+01$
Fig. 6.16(d)	$5.920e-01$	$2.340e+00$	$1.906e-01$	3219	498	$1.418e+03$	$2.581e+02$

Since one iteration of the SIM corresponds to an actual time step, whereas it is a complete simulation of the glacier evolution over some time frame in the SOA, their convergence rates cannot be compared. Therefore, the convergence rates of the TIM and the SOA are confronted. The TIM applied to steady geometries consists in adapting Algorithm 5 to the steady case, where $s_i = s_f = s^{\text{obs}}$ and the same time frame as in the SOA is chosen. Here, the values obtained with the TIM and the fixed-point parameter $\beta = 0.1$ are listed, because the errors are smaller in this case and the convergence rates are not very different from those with larger β . Furthermore, the regularization parameter is set to $\varepsilon = 100$ to make it fast. Table 6.3 compares the results obtained with the SOA to those with the TIM. As before, the regularization parameter r_1 is sought that makes the SOA converge towards a bedrock topography whose discrepancy with the solution bedrock is the same as with the TIM. Again, the computed final surface s_f with the estimated bedrock topography is closer to the exact one if we apply the SOA. In Table 6.3, the number of iterations of the two methods can be compared, since, this time, every iteration consists in simulating the glacier over the chosen time frame. Both the number of iterations and the computational time are much smaller with the SOA. Figure 6.3 superimposes the convergence rates of the two methods and evinces the faster rate of the SOA.

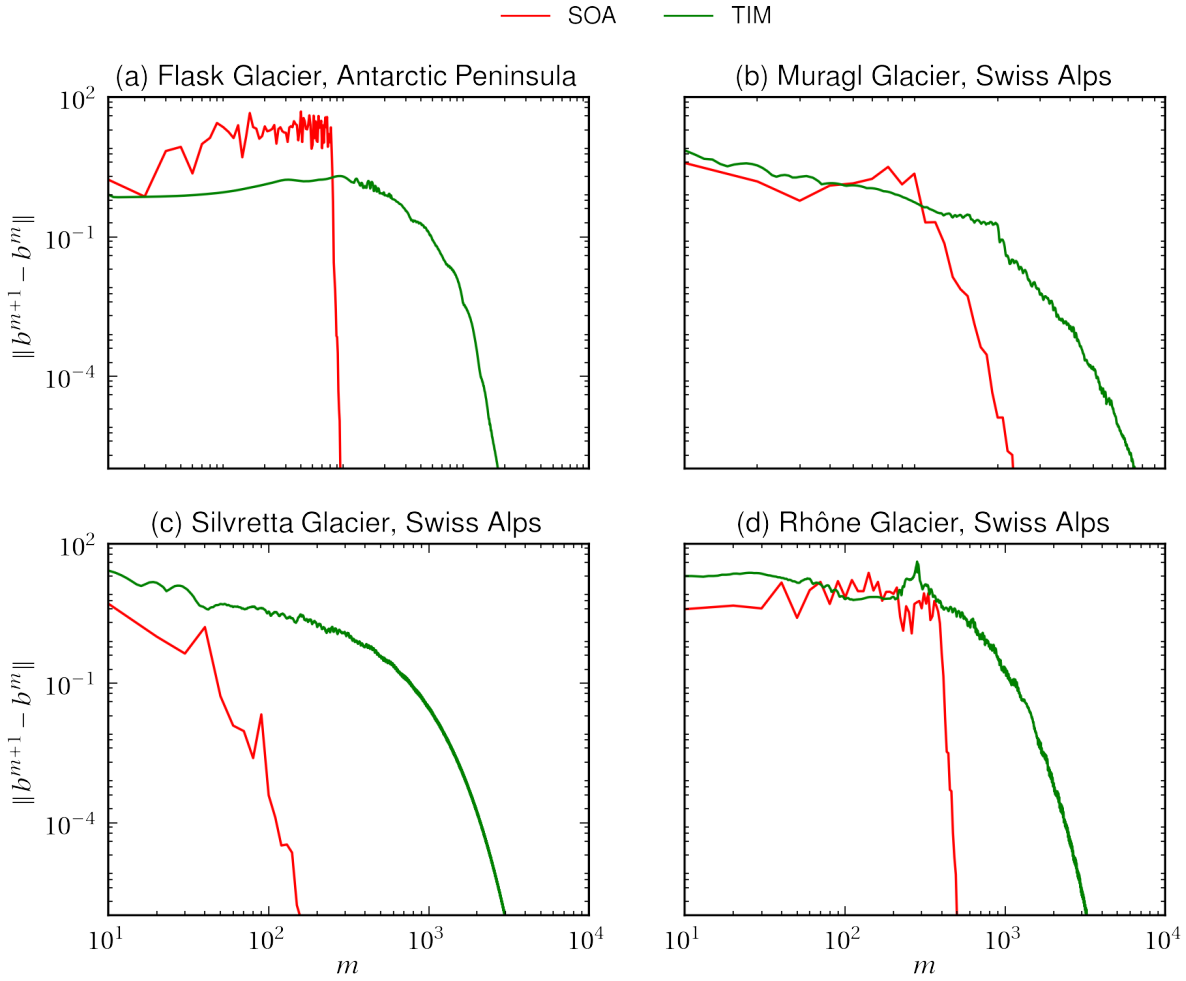


Figure 6.3 – Comparison of the convergence rates of the TIM and the SOA in the case of the steady geometries shown in Figure 6.15, for parameters that lead to about the same error in the estimation of the bedrock topography in both cases (see first two columns of Table 6.3), when the time frame is 1 a. The TIM regularization and fixed-point parameters were set to $\varepsilon = 100$ and $\beta = 0.1$. The applied SOA regularizations are listed in Table 6.3. Convergence is faster with SOA than with TIM.

Transient geometries

The efficiency of the TIM is now compared to that of the SOA in the case of transient geometries. To make this comparison possible, the same time frame must be chosen in both methods and, as above, the regularization parameter r_1 of the SOA must be selected in such a way that the discrepancy $\|b - b^*\|$ is about the same in both cases. Tables 6.4-6.5-6.6 report the discrepancies between the computed and solution bedrock topographies as well as between the computed and exact final surfaces, the number of iterations, and the computational time needed to converge for time frames of 1 a, 5 a, and 10 a when the glaciers' boundaries are fixed. The results are essentially the same as in the steady case. The resulting final surface discrepancy is smaller with the SOA than the TIM. The high values of r_1 applied to geometry of Figure 6.15(c) are due to the fact that a Gaussian filtering was applied to the bedrock topography after each TIM iteration, allowing for convergence but also increasing the error in the estimation. To get these errors with the SOA, a large value of r_1 was used. Because the algorithm is the same as in the steady case, the behavior of the convergence rate is the same.

Table 6.4 – Discrepancy $\|s_f - s^{\text{obs}}\|$ between the computed and the solution final surface geometries s_f and s^{obs} (the discrepancy $\|b - b^*\|$ between the computed and the solution bedrock topographies b and b^* is not listed here but they are the same in the two methods), number of iterations, and computational time needed to converge in both the TIM and SOA applied to a fixed-boundary, transient geometry, when the time frame is 1 a. In the TIM, the regularization and fixed-point parameters are set to $\varepsilon = 100$ and $\beta = 0.1$ respectively. As in the previous experiments, the regularization parameters r_1 are chosen in such a way that the discrepancy $\|b - b^*\|$ is the same in both TIM and SOA. As in the steady case, the quality of the estimation with SOA is better than with TIM.

	r_1	$\ s_f - s^{\text{obs}}\ $ [m]		n_{its}		T_{CPU} [s]	
		TIM	SOA	TIM	SOA	TIM	SOA
Fig. 6.16(a)	$2.502e-01$	$5.920e-01$	$6.379e-03$	17493	846	$1.651e+04$	$1.146e+03$
Fig. 6.16(b)	$6.032e-01$	$6.524e-01$	$9.772e-02$	925	159	$2.190e+02$	$6.628e+01$
Fig. 6.16(c)	$5.506e+00$	$8.587e-01$	$6.064e-01$	950	177	$2.040e+02$	$2.874e+01$
Fig. 6.16(d)	$5.549e-01$	$6.493e-01$	$1.021e-01$	2500	334	$9.263e+02$	$2.008e+02$

Table 6.5 – Same comparison as in Table 6.4 but over a time frame of 5 a.

	r_1	$\ s_f - s^{\text{obs}}\ $ [m]		n_{its}		T_{CPU} [s]	
		TIM	SOA	TIM	SOA	TIM	SOA
Fig. 6.16(a)	$6.502e-01$	$5.996e-01$	$1.029e-02$	17309	420	$6.996e+04$	$2.972e+03$
Fig. 6.16(b)	$4.248e+00$	$6.589e-01$	$3.107e-01$	853	80	$7.076e+02$	$1.334e+02$
Fig. 6.16(c)	$1.000e+02$	$3.024e+00$	$2.860e+00$	611	93	$3.774e+02$	$7.212e+01$
Fig. 6.16(d)	$2.462e-01$	$6.245e-01$	$4.483e-02$	2459	449	$3.458e+03$	$1.142e+03$

Table 6.6 – Same comparison as in Tables 6.4 and 6.5 but over a time frame of 10 a.

	r_1	$\ s_f - s^{\text{obs}}\ $ [m]		n_{its}		T_{CPU} [s]	
		TIM	SOA	TIM	SOA	TIM	SOA
Fig. 6.16(a)	$7.054e-01$	$6.067e-01$	$1.128e-02$	17286	381	$1.483e+05$	$5.369e+03$
Fig. 6.16(b)	$3.489e+00$	$6.237e-01$	$2.406e-01$	797	82	$1.242e+03$	$2.163e+02$
Fig. 6.16(c)	$1.563e+01$	$9.010e-01$	$4.149e-01$	374	64	$3.969e+02$	$1.035e+02$
Fig. 6.16(d)	$2.691e-01$	$6.029e-01$	$4.742e-02$	2444	401	$6.540e+03$	$1.854e+03$

Figures 6.4(a) and (b) depict the estimations in the case of retreating glaciers, while Figures 6.4(c) and (d) those of growing glaciers. Technically, the estimation for expanding glaciers is very easy to do. However, the retreating case may be unstable and harder to deal with. This is because of the negativity of the surface mass-balance at locations where the ice thickness is very small or zero. In this case, much care has to be taken about bound (6.25) so that accurate estimations can be carried out. Table 6.7 reports the errors, numbers of iterations and computational times done in both the TIM and SOA. The same conclusions as before can be reached, *i.e.* the estimated topography is of higher quality with SOA than with TIM, with this set of parameters.

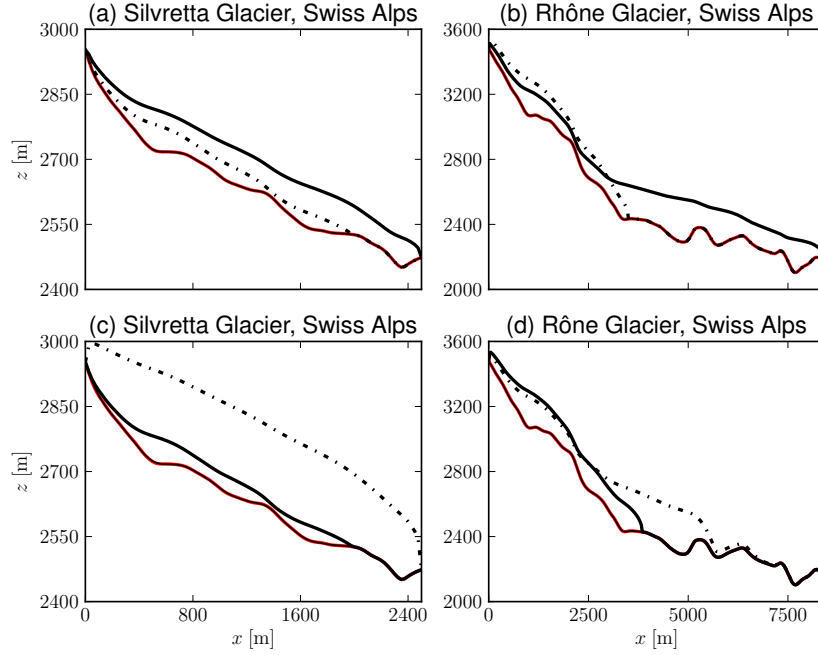


Figure 6.4 – Bedrock topography estimation in the transient case when the glaciers are retreating ((a) and (b)) or expanding ((c) and (d)). The thick continuous and dashed lines represent the initial and final surfaces respectively, while the thick red and thin black lines show the solution and estimated subglacial topographies. For these simulations, we set $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. The time frames are different: (a) $t_f - t_i = 15$ a (b) $t_f - t_i = 5$ a (c) $t_f - t_i = 5$ a (d) $t_f - t_i = 10$ a. Each time, the surface mass-balance is chosen in such a way that a large retreat or advance could be generated. As before, the estimation is very close to the solution topography. Corresponding errors, number of iterations, and computational times are listed in Table 6.7.

Table 6.7 – Discrepancy $\|s_f - s^{\text{obs}}\|$ between the computed and solution final surface geometries s_f and s^{obs} (the discrepancy $\|b - b^*\|$ between the computed and solution bedrock topographies b and b^* are not shown here as they match almost exactly in all cases), number of iterations, and computational times needed to converge in both the TIM and SOA applied to a moving-boundary, transient geometry, over various time frames (listed in the caption of Figure 6.4). In the TIM, we set the regularization and fixed-point parameters to $\varepsilon = 100$ and $\beta = 0.1$ respectively. The regularization parameter r_1 is chosen in such a way that the discrepancy $\|b - b^*\|$ is the same in both the TIM and SOA. The quality of the estimation with SOA is better than with TIM.

	r_1	$\ s_f - s^{\text{obs}}\ $ [m]		n_{its}		T_{CPU}	
		TIM	SOA	TIM	SOA	TIM	SOA
Figure 6.4(a)	$5.716e-01$	$3.196e-02$	$1.858e-02$	2548	49	$3.574e+03$	$1.630e+02$
Figure 6.4(b)	$5.000e+00$	$1.355e+01$	$1.335e-01$	732	107	$1.154e+03$	$3.530e+02$
Figure 6.4(c)	$1.565e+00$	$4.157e-01$	$1.001e-01$	1468	85	$8.220e+02$	$7.000e+01$
Figure 6.4(d)	$9.409e-01$	$5.931e-01$	$5.732e-02$	1146	178	$4.085e+03$	$1.022e+03$

6.3.2 Three-dimensional formulation

To validate Algorithms 7 and 8, a set of bedrock and surface topographies, surface velocities as well as a surface mass-balance time series \mathcal{B} are assumed that satisfy problem (2.31) on the geometries of Silvretta, Gries, and Muragl glaciers with sliding included over a time frame of 10 a. The sliding parameters are set to $z_{\text{sl}} = 2700$ m and $\mathcal{C} = 0.025 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$ on Silvretta and Gries glaciers (colored, contour lines on the large contour plot of Figures 6.7

and 6.8), as in the previous chapter, for the sake of comparison. On Muragl glacier, they are set to $z_{sl} = 2400$ a and $\mathcal{C} = 0.025 \text{ bar}^{-1} \text{ a}^{-\frac{1}{3}}$. In all cases, the surface mass-balance parameters are set to $z_{ELA} = 2800$ m and $m_1 = m_2 = 0.01 \text{ a}^{-1}$, and the rheology parameters are $A = 0.1 \text{ bar}^{-3} \text{ a}^{-1}$ and $n = 3$. The only model parameter that is not subjected to optimal control in the next numerical experiments is Glen's flow law exponent n .

Thereafter, several numerical experiments are presented that validate the three-dimensional shape optimization algorithm presented in Section 6.2. First, all the model parameters are supposed known and the subglacial topography of Silvretta and Gries glaciers is reconstructed from the observation of the surface topography at final time, so as to compare the results to those obtained with the quasi-stationary and transient inverse methods in Sections 5.5.2 and 5.5.3. In this context, the consideration of several surface topographies during the inversion is studied on Muragl glacier. Indeed, the question of the relevance of the incorporation of several surface topography observations is important if they can improve the algorithm's convergence or, to the contrary, overdetermine the problem.

Second, numerical experiments aiming at the reconstruction of the bedrock topography and some model parameters from the knowledge of surface data are presented. The computation of the ice thickness distribution along with the rate factor and the surface mass-balance parameters is performed on Gries and Silvretta glaciers from the observation of the surface topography at final time only. Then, from the observation of various surface topographies and velocities, the ice thickness distribution is determined on Silvretta glacier along with, first, the rate factor, second, the surface mass-balance parameters, and third, the sliding law parameters.

Reconstruction of the subglacial topography

Suppose only the bedrock topography is unknown while all the model parameters are accurately known. Aim is now to reconstruct the latter topography from the final surface topography alone. The time and space steps are here set to $\Delta t = 10^{-2}$ a and $\Delta x = 50$ m respectively and the regularization parameters are $r_1 = r_2 = 0$. The decrease of the surface and bedrock topography misfits is represented on Figures 6.5(a) and (b) in the two cases where $t_f = 1$ and 10 a on the geometry of Silvretta glacier. Compared to Figures 5.8 and 5.12 that depict the results obtained in the same case with the quasi-stationary and transient inverse methods, it is noticeable that both misfits are decreased much more with the shape optimization algorithm. The surface topography misfit reaches a small value of about 10^{-6} m^2 in both time frames, while the bedrock topography misfit reaches about 10^3 m^2 . The surface topography misfit can even be as small as the machine precision in some cases, although the bedrock topography misfit remains significant. In this particular case, the maximal discrepancy between the computed and solution bedrock topographies is about 10 m while it is only 0.02 m between the computed and exact final surfaces. Besides, Figure 6.5(c) shows that the gradient goes to zero with $\frac{1}{\ell^3}$ in both cases, *i.e.* the shape optimization algorithm converges very fast with the number of iterations. The computation taking into account the glacier's 10 a-evolution performs a bit worse, resulting in a higher bedrock topography misfit.

Figure 6.6 illustrates how powerful the inversion method is for shallow ice flows over a time frame of 1 a, compared to its counterparts Figures 5.9 and 5.13 in the quasi-stationary and transient inverse methods respectively. Indeed, all the points defining the ice thickness distribution estimated with the shape optimization algorithm are located very near to the dashed line that marks the points where the estimation coincides with the exact solution. Figures 6.7 and 6.8 show profile curves across the glacier that compare the exact to the estimated solutions. No significant discrepancy can be noticed along these profile lines.

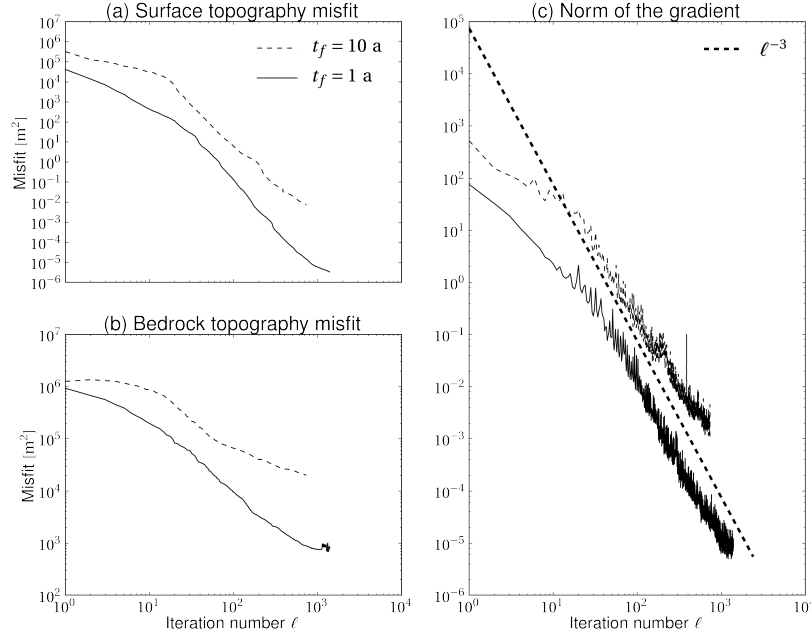


Figure 6.5 – Surface and bedrock topography misfits as functions of the iteration number on Silvretta glacier when the three-dimensional, shape optimization algorithm 8 is applied, over time frames of 1 and 10 a. The simulation are performed with space and time steps $\Delta x = 50$ m and $\Delta t = 10^{-2}$ a respectively and regularization parameters $r_1 = r_2 = 0$. The misfit represented here is the square of the L_2 -norm of the topographies' discrepancy, namely (6.33). Compared to the results obtained with the quasi-stationary and transient inverse methods (see Figures 5.11 and 5.12), both misfits are several orders of magnitude smaller. Here, the maximum deviation in the bedrock topography is 10 m, while it is only 0.02 m in the final surface topography.

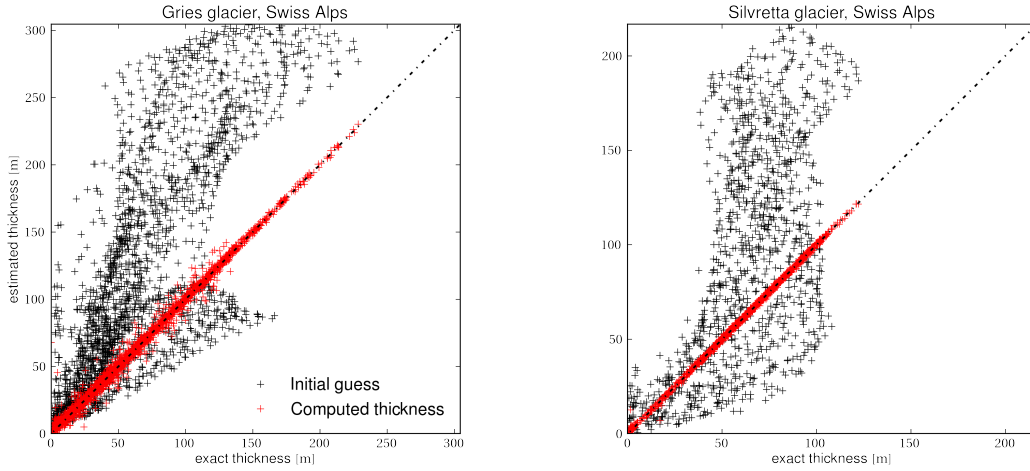


Figure 6.6 – Ice thickness comparisons on Gries (left) and Silvretta (right) glaciers. A time evolution over 1 a is considered. The space and time steps are set to $\Delta x = 50$ m and $\Delta t = 10^{-2}$ a respectively and the regularization parameters to $r_1 = r_2 = 0$. The black markers depict the comparison of the thickness' initial guess to the exact ice thickness, while the red markers show the the comparison of the computed ice thickness with the shape optimization algorithm 8 to the exact thickness. Note the difference of scales. The results are much better than those obtained with the quasi-stationary and transient inverse methods (see Figures (5.9) and (5.13)). The algorithm solves the inverse problem almost perfectly, as most of the red markers lie exactly on the dashed line, which represents the points such that the exact thickness equals the computed thickness.

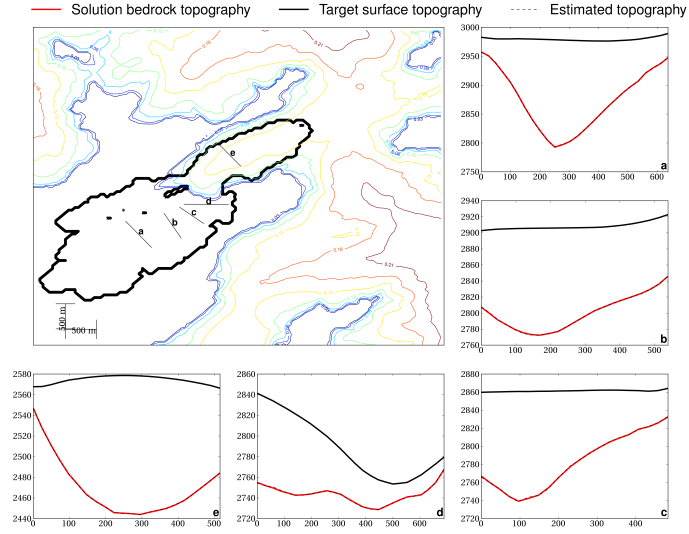


Figure 6.7 – Illustration of the bedrock topography estimated with Algorithm 8 from a 1 a-evolution of Gries glacier that complements the global Figure 6.6(a). The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot.

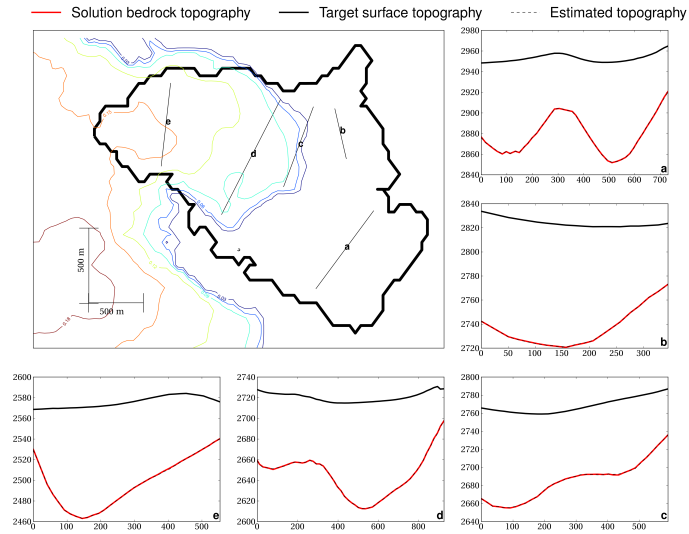


Figure 6.8 – Illustration of the bedrock topography estimated with Algorithm 8 from a 1 a-evolution of Silvretta glacier that complements the global Figure 6.6(b). The large contour plot shows the location of the selected profile lines across the glacier (black lines) as well as the sliding domain (the colored contour lines give $\frac{1}{C}$, expression (2.10)). The small subplots show the estimation along each of the profile lines. Each time, the red curve stands for the solution bedrock topography, the dashed curves for the estimated topographies (final surface and bedrock), and the black curve for the target surface topography. Note the scale difference on each subplot.

Finally, Figure 6.9 shows how pointless it is to incorporate several observed surface topographies in the computation

of the bedrock topography on the geometry of Muragl glacier. Three different simulations are run, where the observed surface topography either at time $t = 10$ a, or at times $t = 8$ and 10 a, or at times $t = 5, 8$, and 10 a are taken into account during the computation, as explained in the end of Section 6.2.1. The decrease of both the cost functional and its gradient is represented on the Figure. The behavior of the three curves is essentially the same, *i.e.* none of the inversions is done faster. Moreover, the additional surface topography information does not improve the result. Even worse, it may overdetermine the problem and possibly empede the algorithm's convergence.

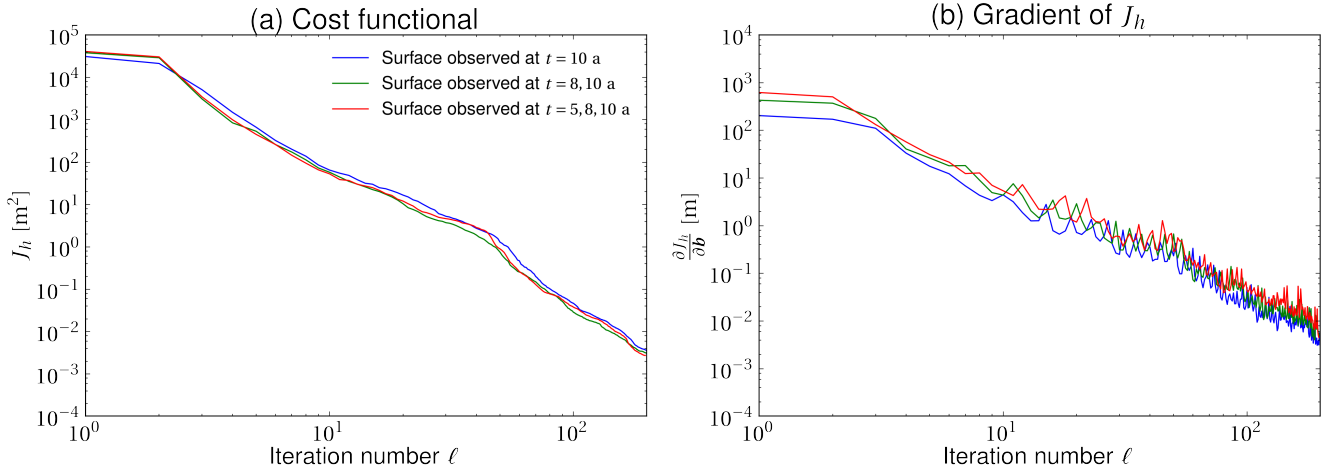


Figure 6.9 – Cost functional and its gradient as functions of the iteration number on Muragl glacier when the shape optimization algorithm 8 takes into account the surface topography observed at time $t = 10$ a (blue curve), at times $t = 8$ and 10 a (green curve), and at times $t = 5, 8$, and 10 a (red curve) (*i.e.* the Algorithm is modified with (6.64)–(6.66)). The behavior of the algorithm is the same in the three cases, hence the incorporation of any other surface topography information than the final surface topography is pointless.

Simultaneous reconstruction of the subglacial topography and the model parameters

Assume now that in addition to the bedrock topography, both the rheology and the surface mass-balance parameters are unknown. Aim is now to recover them all from the observation of the final surface topography only, when the time frame is 1 a. As before, suppose $m_1 = m_2$ in the expression (2.34) of the surface mass-balance. The results of this inversion on Silvretta and Gries glaciers are illustrated by Figures 6.10, 6.11, and 6.12. Figure 6.10 represents the evolution of the cost functional, the rate factor, and the surface mass-balance parameters with the iteration number, while Figure 6.11 reports the evolution of the corresponding gradients. On Figures 6.10(b)-(d), the dashed, black lines indicate the values that have to be recovered. Figure 6.12 compares the estimated to the exact ice thicknesses in both cases. Figure 6.10(a) shows that the algorithm succeeded minimizing the cost functional. In fact, the maximal discrepancy between the exact and computed final surface topographies is about 3 m, while it is about 15 m between the estimated and exact bedrock topographies. Hence, apparently, the algorithm performed well. However, Figures 6.10(b) and (c) report that the rate factor and the surface mass-balance's equilibrium line altitude are not recovered precisely, contrary to the melting rate m_1 (Figure 6.10(d)). This phenomenon occurs because the model is not sensitive enough to these parameters, as shown by Figure 6.11, which shows how small the values of the the cost functional's derivatives are with respect to each of the parameters. Moreover, it also shows how different in magnitude they are. Nevertheless, the estimation of the bedrock topography is good enough even though not all the model parameters are correct. It is still better than those obtained with the quasi-stationary and transient inverse methods with the right parameters.

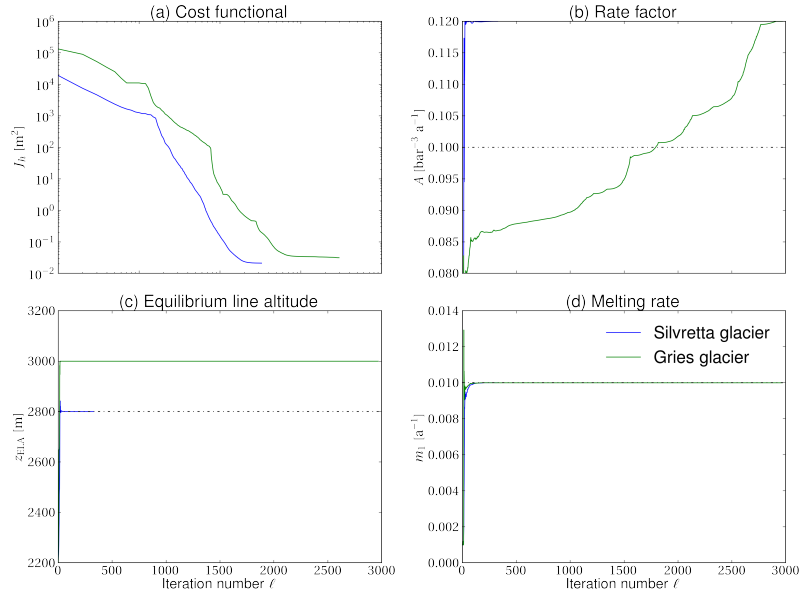


Figure 6.10 – Simultaneous estimation of the subglacial topography, the rate factor A , and the surface mass-balance parameters z_{ELA} and $m_1 = m_2$ from the observation of the final surface topography only over a time frame of 1 a on the geometries of Silvretta (blue curves) and Gries (green curves) glaciers. The true value of each parameter is indicated with the black, dash-dotted lines, namely $A = 0.1 \text{ bar}^{-3} \text{ a}^{-1}$, $z_{ELA} = 2800 \text{ m}$, and $m_1 = m_2 = 0.01 \text{ a}^{-1}$. The algorithm is not able to retrieve the rate factor and the equilibrium line altitude, although the cost functional reaches a sufficiently small value (about 10^{-2} m^2).

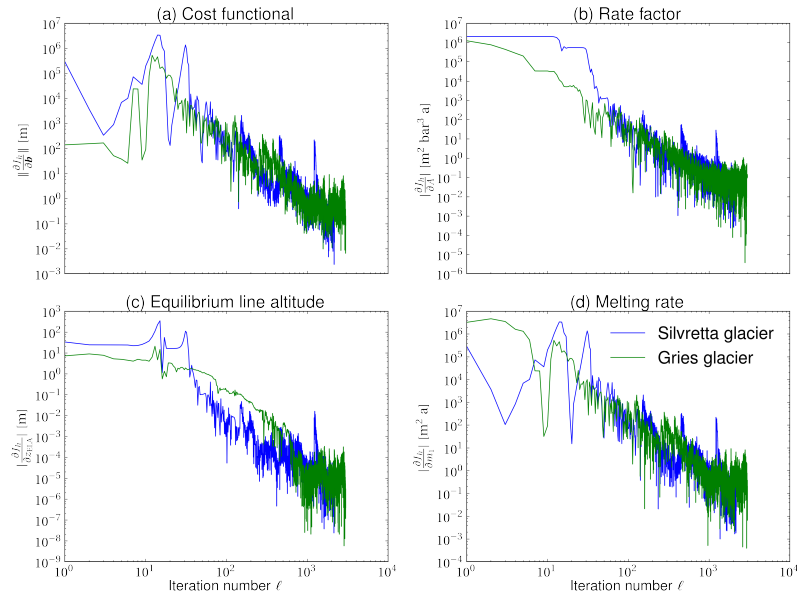


Figure 6.11 – Norm of the gradients of the cost functional shown on Figure 6.10 with respect to the bedrock topography, the rate factor, and the surface mass-balance parameters on the geometries of Silvretta (blue curves) and Gries (green curves) glaciers. Although the algorithm does not retrieve the right values of the model parameters, the gradient of J_h with respect to these parameters is very small.

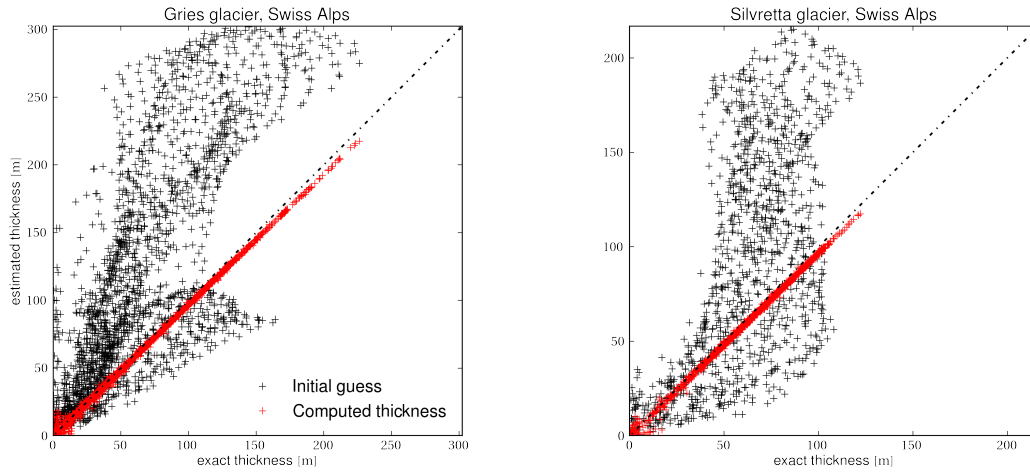


Figure 6.12 – Ice thickness comparisons on Gries (left) and Silvretta (right) glaciers when the bedrock topography is inferred in the same time the surface mass-balance and rheology parameters are calculated. A time evolution over 1 a is considered. The space and time steps are set to $\Delta x = 50$ m and $\Delta t = 10^{-2}$ a respectively and the regularization parameters to $r_1 = r_2 = 0$. The black markers depict the comparison of the thickness' initial guess to the exact ice thickness, while the red markers show the comparison of the computed ice thickness with the shape optimization algorithm 8 to the exact thickness. Note the difference of scales. Because the rheology parameter cannot be inferred with precision in these cases, the ice thickness is recovered with a lower precision. The estimation is still good enough.

If the sliding law parameters are unknown, in addition to the bedrock topography, the rheology, and the surface mass-balance parameters, then the recovery of the subglacial topography from the observation of the surface topography at final time is similar to that of the previous numerical experiment. Again, the algorithm is not capable of determining all the parameters, especially the sliding law parameters, although it actually converges and the cost functional attains a sufficiently small value, of the order of 10^{-2} m². As mentioned above, this is because the model is not very sensitive to the sliding parameters. Section 6.4.2 gives more details about this sensitivity problem.

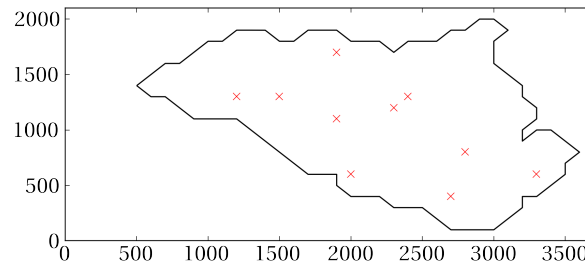


Figure 6.13 – Sample surface ice velocity data (red markers) on Silvretta glacier used for the conjoint computation of the subglacial topography and some model parameters.

Finally, from the knowledge of various surface data over a time frame of 5 a, the subglacial topography is computed on the geometry of Silvretta glacier together with, first, the rate factor, second, the surface mass-balance parameters, and third, the sliding law parameters. In a first experiment, they are determined from the surface topography at final time. In a second experiment, they are calculated from the surface topography at times 3 and 5 a. In a last experiment, only the surface topography and a partial grid of surface ice velocities (see Figure 6.13) are taken into account in the inversion. Figure 6.14 reports the results of the numerical experiments. Each subplot (a)–(e) represents the evolution of each of the studied parameters with the iteration number. Each colored curve represents

a different experiment, where different surface data are used during the inversion. The black, dashed line indicates the parameter's value that has to be recovered. Figure 6.14(f) depicts the evolution of the cost functional with the iteration number in all experiments. As far as one can see, the algorithm struggles to determine the sliding law parameters, whereas it performs well for the computation of all the other model parameters, especially when the surface ice velocities are taken into account. As in the previous experiments, the cost functional takes a very low value, although some parameters cannot be found accurately.

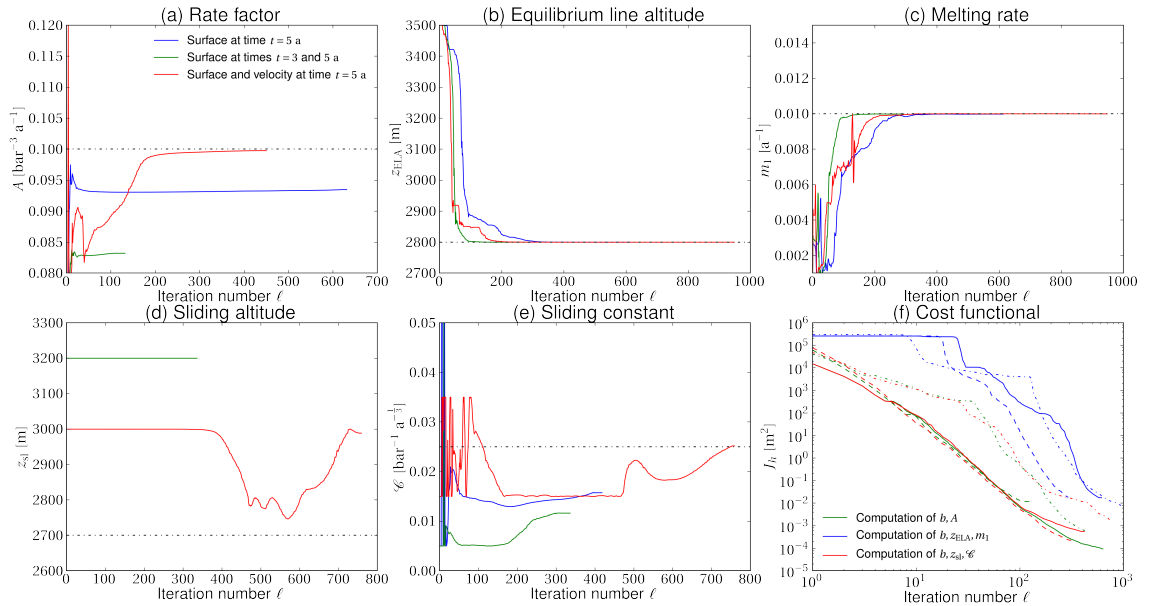


Figure 6.14 – Convergence of the shape optimization algorithm in the three cases where only the surface topography at final time $t_f = 5$ a (blue curves in plots (a)-(e), continuous curves in plot (f)), the surface topographies at times $t = 3$ and 5 a (green curves in plots (a)-(e), dashed curves in plot (f)), and both the surface topography and ice velocity at final time $t_f = 5$ a (red curves in plots (a)-(e), dash-dotted curves in plot (f)) are used on the geometry of Silvretta glacier. These results correspond to three different numerical experiments, namely the conjoint computation of both the bedrock topography and the rate factor (plot (a) and green curves in plot (f)), the conjoint computation of both the bedrock topography and the surface mass-balance parameters (plots (b)-(c) and blue curves in plot (f)), and the conjoint computation of both the bedrock topography and the sliding law parameters (plots (d)-(e) and red curves in plot (f)). The best results are obtained when the surface ice velocity is taken into account. Again, the algorithm has more difficulties estimating the sliding parameters.

To conclude, let us mention that a dissociated algorithm was also applied. All the parameters are first set to some initial guess, as the bedrock topography. Then, the method consists in the iterative computation with Algorithm (8) of, first, the bedrock topography alone with the previously determined values of the model parameters, second, the rate factor with the previously computed bedrock topography and the other model parameters, third, the surface mass-balance parameters with the previously computed bedrock topography, rate factor, and sliding parameters, and, last, the sliding parameters with the previously computed bedrock topography, rate factor, and surface mass-balance parameters. After each iteration of the method, every parameter is set a new value separately. None of the numerical experiments performed with this method on various geometries supplied better results than those described above in this section.

6.4 Sensitivity analysis

A short sensitivity analysis is now proposed for the algorithms introduced in this Chapter, first in two space dimensions, then in three space dimensions. Essentially, the sensitivity of the methods to the regularization parameters and surface data is investigated. In the three-dimensional case, the sensitivity of the forward model to the surface mass-balance and sliding law parameters is also addressed, in order to explain why the shape optimization algorithm cannot compute all these parameters in the same time.

6.4.1 Two-dimensional algorithm

Let us first apply the SOA with various parameters r_1 to the four glacier geometries studied in the previous chapter. Figure 6.15 shows the resulting estimations when the time frame is 1 a. Most of the time, the large value $r_1 = 10$ leads to estimations too far from the solution, while the other ones are close to it. When r_1 is small enough, no large discrepancy can be distinguished between the estimated subglacial topography and the solution bedrock.

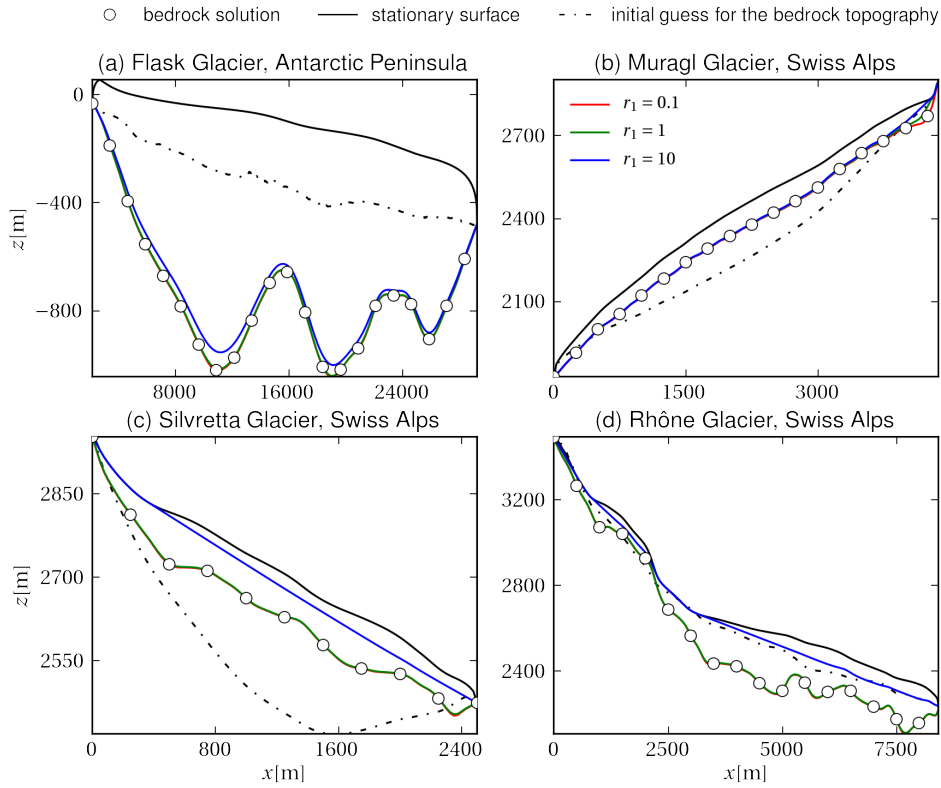


Figure 6.15 – Various examples illustrating the application of Algorithm 6 to steady geometries, with $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, $\delta = 10^{-6}$, and $t_f - t_i = 1$ a for different values of the regularization parameter r_1 (red, green, and blue lines). The continuous black line represents the stationary surface and the dashed line the initial guess b^0 for the bedrock topography. The white circles delineate the solution bedrock. When r_1 is small, the computed and the exact solution are hard to distinguish. Errors in the bedrock and final surface topography estimations are listed in Table 6.1 for $r_1 = 1$ and various time frames. Table 6.2 lists the errors, computational times, and numbers of iterations when r_1 is such that the error in the basal estimation is comparable to the one obtained with SIM for each geometry.

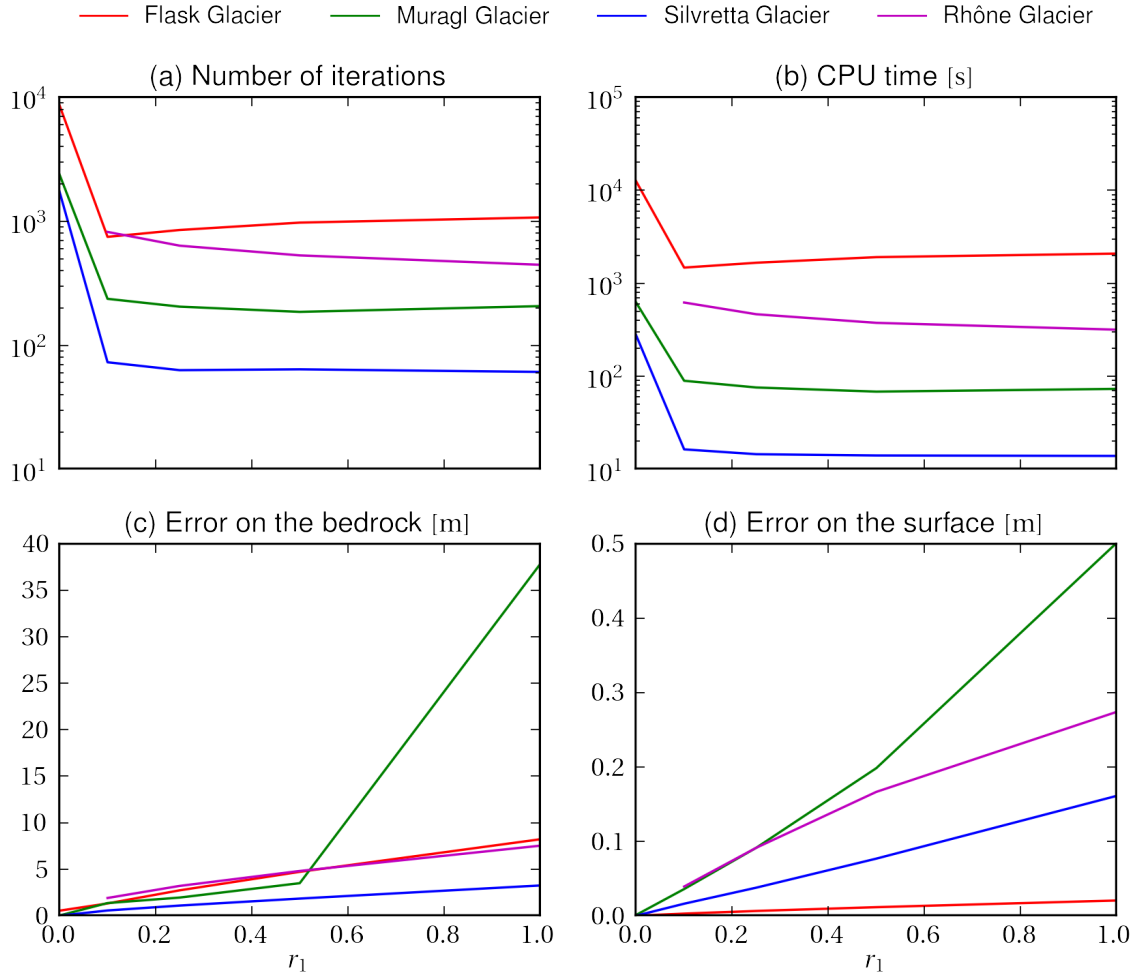


Figure 6.16 – Sensitivity of the method to the regularization parameter when $\Delta x = 25$ m, $\Delta t = 10^{-3}$ a, $\delta = 10^{-6}$ m, and a time frame of 1 a for steady geometries. Variations in (a) number of iterations, (b) computational time, (c) error on the bedrock topography, and (d) error on the computed surface from the estimated bedrock topography are shown as a function of r_1 . The number of iterations and the computational time are directly proportional. The errors grow with the regularization parameter. Differences between the various geometries can be large and, already in the range we chose r_1 , the error in the subglacial estimation can be large (e.g. on Muragl glacier). Errors on the bedrock geometry are about one order of magnitude larger than the errors on the surface topography.

Figure 6.16 shows how sensitive the SOA is to the regularization parameter r_1 . Figure 6.16(a) and (b) evince that it is not useful to increase r_1 in order to decrease the computational time. Furthermore, the error in the subglacial topography due to the increase of r_1 is so large that this parameter should be kept as small as possible, although the error in the final surface remains one order of magnitude smaller (see Figures 6.16(c) and (d)), indicating a low response of the surface topography to large fluctuations in the bedrock geometry.

Now, as in Section 5.6.1, let us investigate the effects of surface topography and mass-balance perturbations of the form $a\chi(x)$, where the perturbation's amplitude $a > 0$ is of the order of magnitude of typical measurement errors and $\chi(x)$ is a uniformly distributed random variable on the interval $[-1, 1]$. First $a\chi(x)$ is added to the surface topography and then to the surface mass-balance. When $a = 1$ m, Figure 6.17 shows how sensitive the estimation is to the small perturbation of the initial and final surfaces and how the increase of the regularization parameter affects it. None of the estimations converged. Instead, they oscillated around the solution. However, the smoothness of the estimated topography is improved with increasing r_1 , even though the estimation still differs from the solution by

more than 1 m. To make the algorithm converge, one has to smooth the perturbed data. On the other hand, the SOA is not sensitive to noise in the surface mass-balance, just like the SIM and TIM.

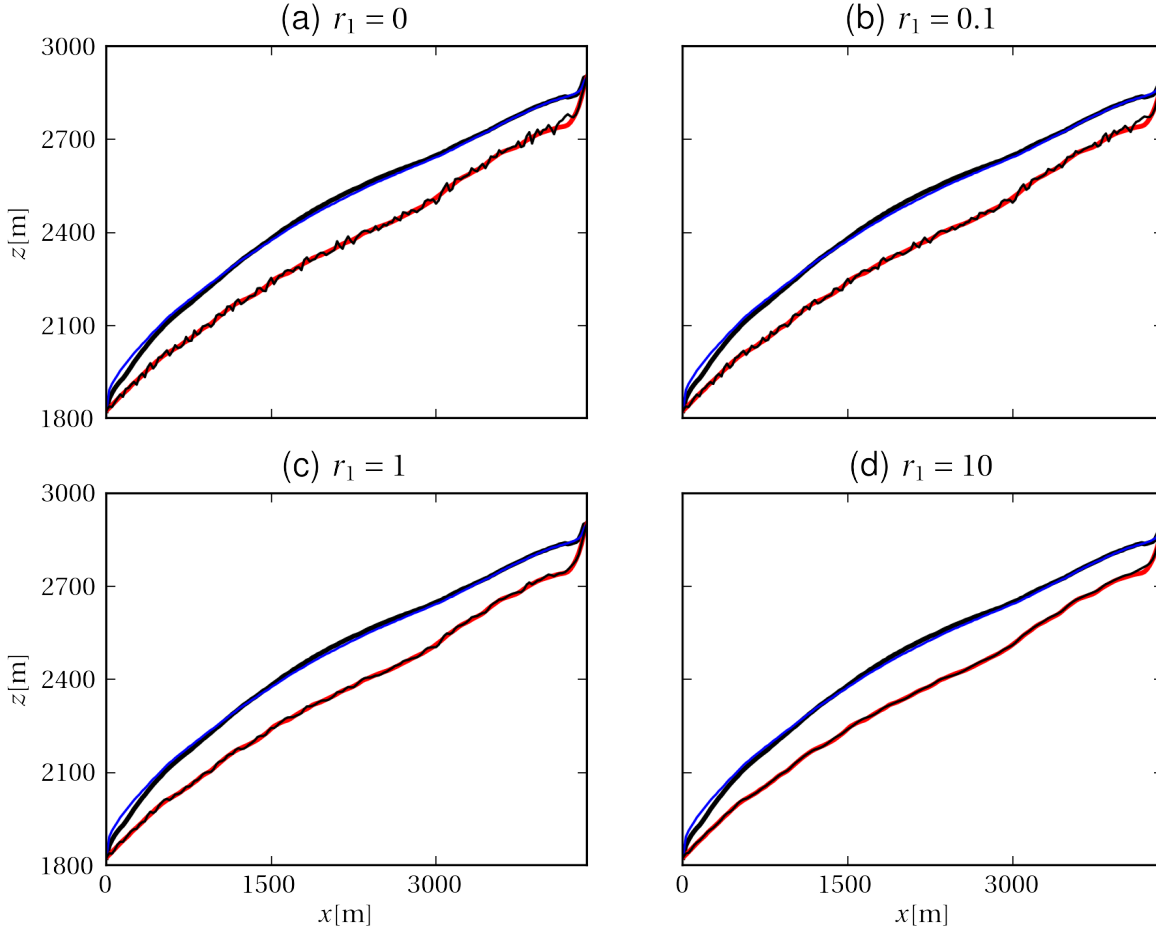


Figure 6.17 – Sensitivity to the surface of Muragl glacier, on a transient, fixed-boundary problem over a time frame of 1 a. We applied the perturbation with $a = 1$ m to both the initial and final surfaces, represented by the thick black and thin blue lines respectively. The perturbation can hardly be noticed on the two surface geometries. The solution and estimated bedrock topographies are delineated by the thick red and thin black lines. Without smoothing of the perturbed data, the algorithm does not converge, but oscillates around the displayed estimation. With larger regularization parameters, the estimation becomes smoother and closer to the solution topography.

6.4.2 Three-dimensional algorithm

The effects of the regularization parameters on the subglacial topography estimation are analogous to those reported in the two-dimensional case above. Of interest in the three-dimensional case are the effects of the gradient and Laplacian regularizations when the surface topography is subjected to measurement errors. As before, perturbations of the form $a\chi(x, y)$ are considered, where the perturbation's amplitude $a > 0$ is of the order of magnitude of typical measurement errors and $\chi(x, y)$ is a uniformly distributed random variable on the interval $[-1, 1]$. Figures 6.18 and 6.19 illustrate the obtained bedrock topography estimations when $a = 5$ m and for various regularization parameters, when only the gradient regularization ($r_2 = 0$) and only the Laplacian regularization ($r_1 = 0$) respectively are applied. Everytime, the surface topography at final time is recovered with a small error compared to that done on the bedrock topography, as in the two-dimensional case. While the surface topography discrepancy is of the order

of about 10 m, the subglacial topography discrepancy is closer to 50 m. Hence, the shape optimization algorithm is very sensitive to the surface topography data, as are all the methods presented in this thesis.

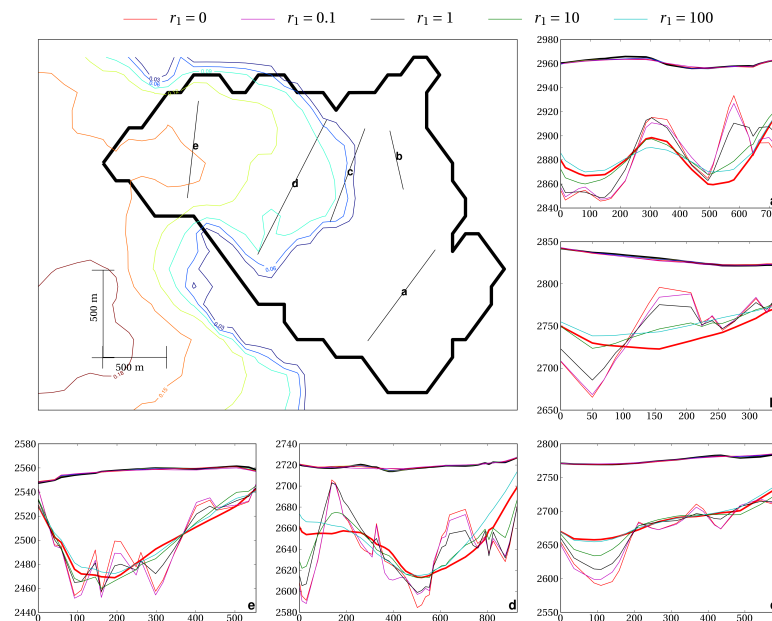


Figure 6.18 – Sensitivity of the shape optimization algorithm 8 when only the gradient regularization applies ($r_2 = 0$) on the geometry of Silvretta glacier when both the initial and the final surface topographies contain a randomly distributed error of 5 m. The time frame is 10 a, and the space and time steps are set to $\Delta x = 100$ m and $\Delta t = 0.1$ a.

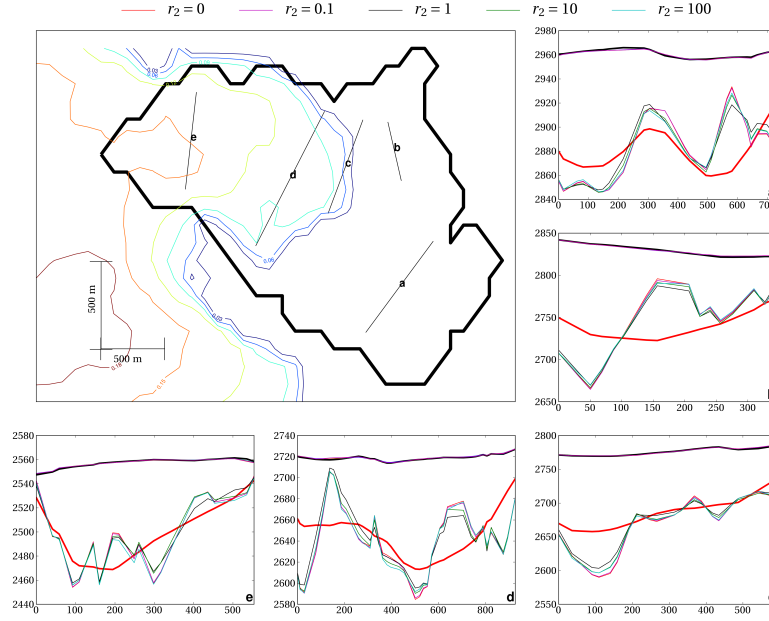


Figure 6.19 – Sensitivity of the shape optimization algorithm 8 when only the Laplacian regularization applies ($r_1 = 0$) on the geometry of Silvretta glacier when both the initial and the final surface topographies contain a randomly distributed error of 5 m. The time frame is 10 a, and the space and time steps are set to $\Delta x = 100$ m and $\Delta t = 0.1$ a.

To conclude the sensitivity analysis of the three-dimensional method, let us show the shape of the cost functional as a function of the surface mass-balance (Figure 6.20(a)) and the sliding law parameters (Figure 6.20(b)) on the geometry of Muragl glacier. These two subplots are generated from the simulation of the forward problem with several values of the surface mass-balance and sliding law parameters and the exact bedrock topography followed by the evaluation of the cost functional (6.33). The other subplots result from the evaluation of its derivative with respect to the displayed parameters. Figure 6.20(c) represents the derivative of J_h with respect to the melting rate m_1 for a fixed equilibrium line altitude ($z_{ELA} = 2800$ m). Subplot (d) depicts the derivative of J_h with respect to the sliding parameter \mathcal{C} for a fixed sliding altitude, subplot (e) that with respect to z_{ELA} for a fixed melting rate, and subplot (f) the derivative with respect to z_{sl} for a fixed value of \mathcal{C} . From Figure 6.20(b), it is clear that the minimization of the functional J_h with respect to the sliding parameters is ill-posed. Indeed, the region where $J_h \approx 0$ is very large. This is essentially due to the term $\vartheta(z_{sl} - b)$ in the formulae (6.76)–(6.79) of the corresponding gradients, which are depicted by subplots (d) and (f). From these plots, it is obvious that there are many solutions that satisfy the condition of a zero gradient. In the case of the surface mass-balance parameter estimation, such a problem does not occur on this geometry, as shown by Figures 6.20(c) and (e). The two-parameter expression of the sliding law proposed in this thesis therefore ends up being inappropriate for optimal control computations. A formulation where the sliding constant could be tuned everywhere in the map domain is preferable, but cannot be applied to Swiss Alps glaciers, as the amount of surface velocity data is too small. Besides, the adimensionalization of Lagrangian (6.36) may provide comparable gradients for each parameter. However, the adimensionalization usually involves some control variables, making this option complicated to apply. For example, a popular choice of ice thickness adimensionalization is through its division by z_{ELA} . Such a process changes completely the Lagrangian functional and its derivatives. Moreover, there is no guarantee that the Algorithm will work better after these modifications. This idea is not developed further here.

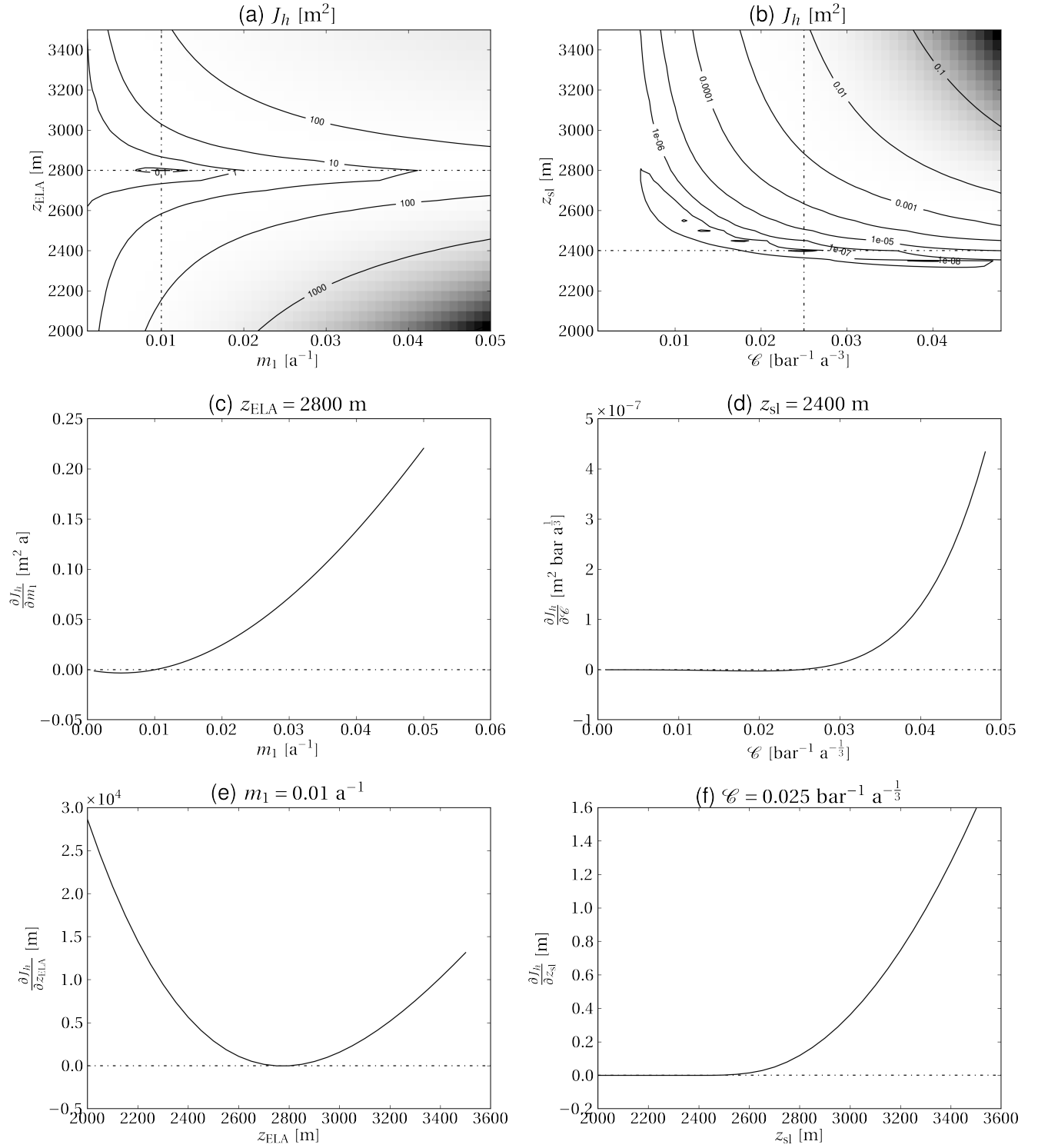


Figure 6.20 – Bivariate cost functionals and corresponding gradients on the geometry of Muragl glacier. (a) Cost functional as a function of the surface mass-balance parameters. (b) Cost functional as a function of the sliding parameters. (c) Gradient of \mathcal{J} with respect to m_1 as a function of m_1 when the equilibrium line altitude is fixed to $z_{\text{ELA}} = 2800$ m and $m_2 = m_1$. (d) Gradient of \mathcal{J} with respect to \mathcal{C} when the sliding altitude is fixed to $z_{\text{sl}} = 2400$ m. (e) Gradient of \mathcal{J} with respect to z_{ELA} when the melting rate is fixed to $m_1 = m_2 = 0.01$ a⁻¹. (f) Gradient of \mathcal{J} with respect to z_{sl} when the sliding coefficient is fixed to $\mathcal{C} = 0.025$ bar⁻¹ a^{-1/3}.

6.5 Application to real-world data

The algorithm is here applied to surface topography measurements collected on Silvretta glacier. The same parameters as before are applied, namely the rate factor (5.38) and the surface mass-balance parameters (5.39). No sliding is assumed. The results are reported on Figures 6.21 and 6.22. Even though Silvretta glacier is the thinnest of the set of studied glaciers, the results are good but not excellent. The algorithm did not converge on this data set, hence the oscillations in the bedrock topography, even with the regularization $r_1 = 0$, $r_2 = 1$. Not represented here, the estimation on Gries glacier is of the same kind. As in the previously shown sensitivity computations of Section 6.4.2, the error on the surface topography is small compared to that on the bedrock topography (Figure 6.22).

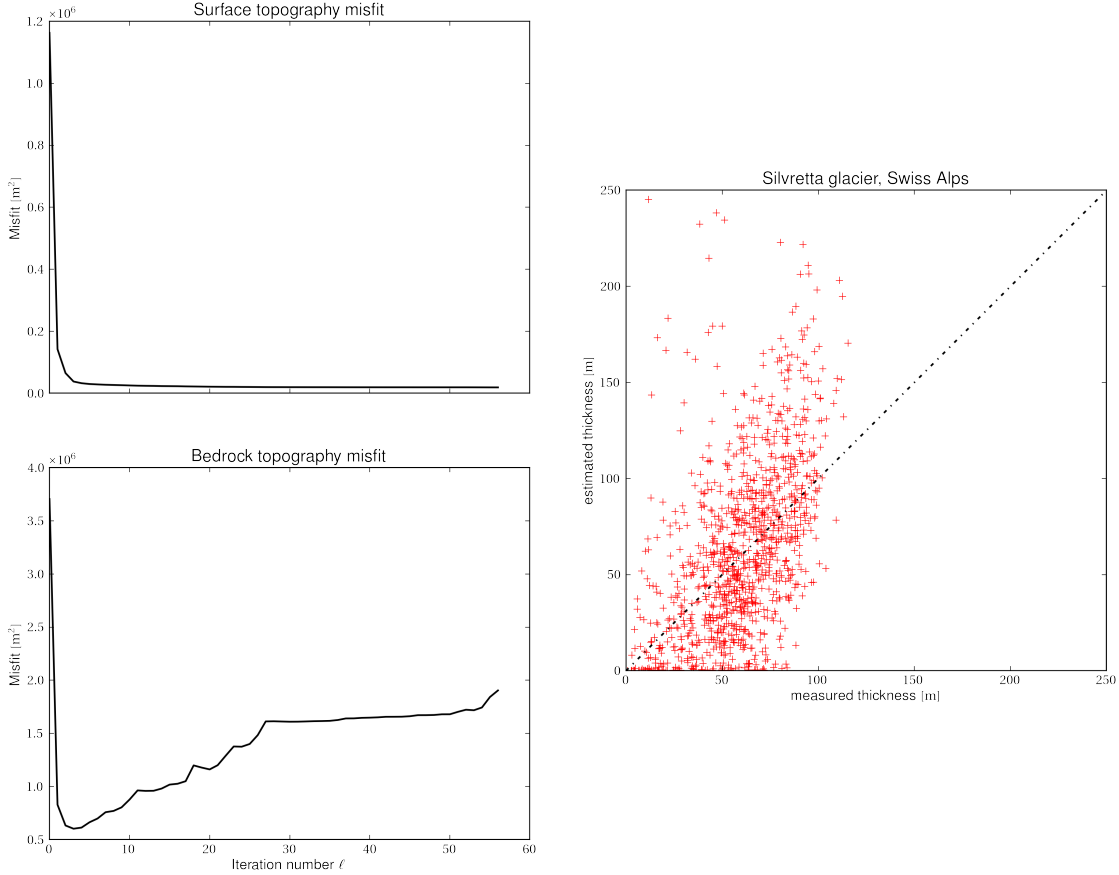


Figure 6.21 – Surface and bedrock topography misfits as functions of the iteration number (left) and resulting bedrock topography estimation after the 56-th iteration of the shape optimization Algorithm 8 on Silvretta glacier, Swiss Alps, for space and time spacings $\Delta x = 50$ m and $\Delta t = 10^{-2}$ a respectively, surface mass-balance parameters (5.39), and no sliding (right). The surface topography misfit decreases during the whole process, contrary to the bedrock topography misfit.

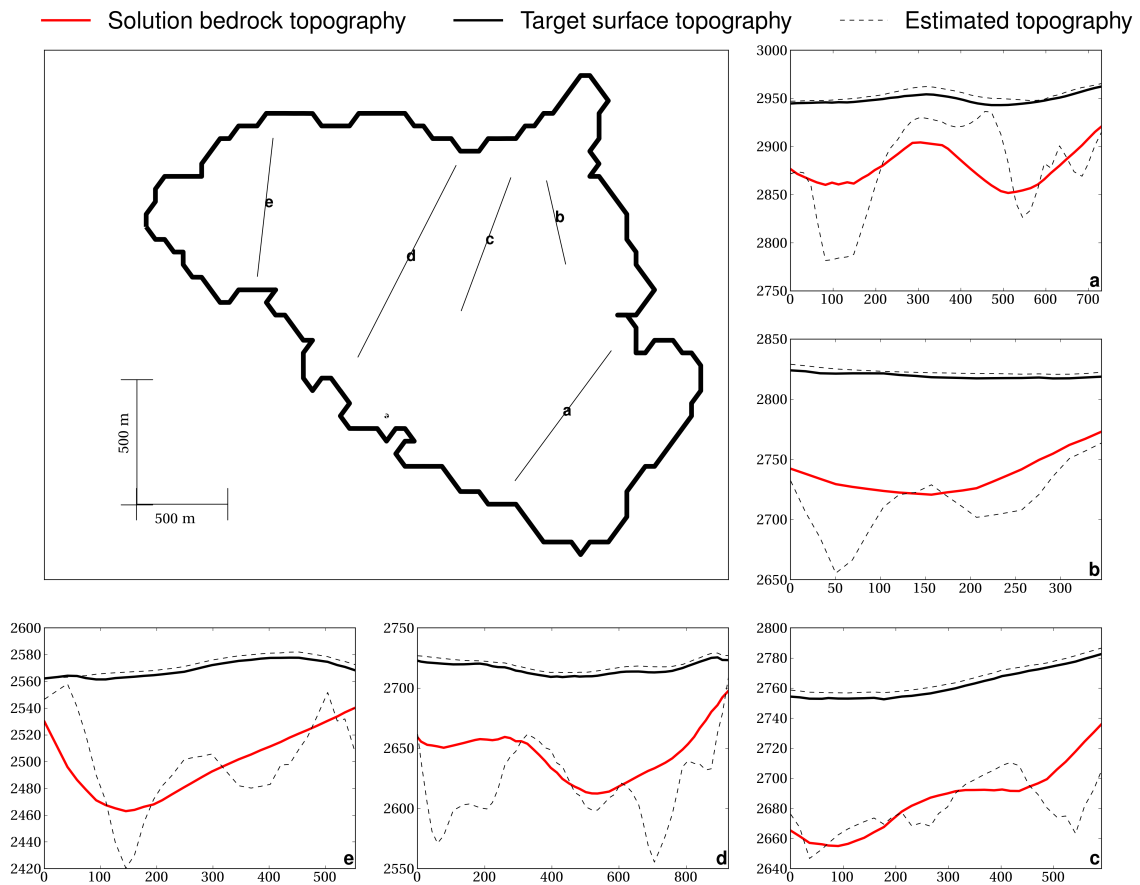


Figure 6.22 – Bedrock topography estimation along profile lines across Silvretta glacier that correspond to the estimation displayed by Figure 6.21. On each subplot, the measured bedrock topography is delineated by the thick, red curve, the target surface topography by the thick, black curve, and the estimated topographies by the thin, dashed curves.

6.6 Perspectives

In order to minimize a functional by means of optimal control, an approximation of its gradient may sometimes be enough. Hence, one may wonder whether the shallow ice gradient contains sufficient information for the minimization of the misfit functional evaluated with the Stokes model. The results of the numerical experiments performed in this thesis to check this feature are not represented here, as they are not good enough. Indeed, the shallow ice gradient proved itself useless for the computation of the bedrock topography and in some cases more or less relevant for the determination of the surface mass-balance parameters. Another outcome of these first experiments is that the procedure seems to be dependent on the mesh size, which is not surprising, as the discretized shallow ice dual problem is solved instead of the discretized Stokes ice dual problem. No convergence can reasonably be achieved in this case.

Nevertheless, the shape optimization algorithm developed in this thesis for shallow ice flows can be extended to Stokes ice flows, *i.e.* the Stokes ice dual problem and corresponding gradient can be derived. For the sake of simplicity, focus is here put on the computation of the subglacial topography from the observation of the final surface topography on a non-sliding glacier. Moreover, the optimization algorithm is developed at the continuous

level only, for the sake of simplicity. In this case, its formulation is simply: *minimize the cost functional*

$$J(b) = \mathcal{J}(\mathcal{H}(b), b) = \frac{1}{2} \int_{\Omega_{\perp}} \left(b + \mathcal{H} \Big|_{t=t_f} - s^{\text{obs}} \right)^2 + \mathcal{R}(b) \quad (6.103)$$

under the constraints (see equation (2.23))

$$\left\{ \begin{array}{lll} \frac{\partial \mathcal{H}}{\partial t} + \nabla \cdot \mathbf{Q} & = \mathcal{B}, & \text{in } \Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} & \geq 0, & \text{in } \Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} & = 0, & \text{on } \partial\Omega_{\perp} \times [t_i, t_f] \\ \mathcal{H} & = s_i - b, & \text{in } \Omega_{\perp} \times \{t_i\}, \end{array} \right. \quad (6.104)$$

and (see equation (2.8))

$$\left\{ \begin{array}{lll} -2\nabla \cdot (\eta D) + \nabla p & = \rho \mathbf{g}, & \text{in } \Omega(b, \mathcal{H}) \\ \nabla \cdot \mathbf{u} & = 0, & \text{in } \Omega(b, \mathcal{H}) \\ -p\mathbf{n} + 2\eta D \cdot \mathbf{n} & = 0, & \text{on } \Gamma_s(b, \mathcal{H}) \\ \mathbf{u} & = 0, & \text{on } \Gamma_b(b, \mathcal{H}), \end{array} \right. \quad (6.105)$$

where Γ_s is the ice–air interface, Γ_b the ice–lithosphere interface (see Figure 2.3), Ω_{\perp} the glacier’s map domain, and $\Omega(b, \mathcal{H})$ the glacier’s ice volume. The glacier’s map domain is independent of both the control and the state variables, contrary to the ice volume. The Lagrangian functional associated to this problem writes

$$\begin{aligned} \mathcal{L}(\mathbf{u}, p, \mathcal{H}, \lambda, \mu, \boldsymbol{\psi}, \chi, b) &= \mathcal{J}(\mathcal{H}, b) + \int_{t_i}^{t_f} \int_{\Omega_{\perp}} \left[\frac{\partial \mathcal{H}}{\partial t} \lambda - \mathbf{Q} \cdot \nabla \lambda - \mathcal{B} \lambda \right] d\mathbf{x} dt + \int_{\Omega_{\perp}} \left(\mathcal{H} \Big|_{t=t_i} - s_i + b \right) \mu d\mathbf{x} \\ &+ \int_{t_i}^{t_f} \int_{\Omega(b, \mathcal{H})} \left[2\eta D(\mathbf{u}) : D(\boldsymbol{\psi}) - p \nabla \cdot \boldsymbol{\psi} - \mathbf{f} \cdot \boldsymbol{\psi} \right] d\mathbf{x} dt + \int_{t_i}^{t_f} \int_{\Omega(b, \mathcal{H})} \nabla \cdot (\mathbf{u}) \chi d\mathbf{x} dt, \end{aligned} \quad (6.106)$$

where the first two integrals correspond to the weak form of constraint (6.104) and the last two integrals to that of constraint (6.105) (see Section 3.2.1). The essential, formal difference with the shallow ice Lagrangian (6.36) is that the domain of integration of some of the arising integrals depend on the control b and state variable \mathcal{H} .

The solutions of the optimization problem (6.103)–(6.105) are among the stationary points of \mathcal{L} . The stationarity of \mathcal{L} with respect to the costate variables λ , μ , $\boldsymbol{\psi}$ and χ implies the constraint equations (6.104) and (6.105). Its stationarity with respect to the state variables \mathbf{u} , p and \mathcal{H} gives the dual problem. Technically, the shape derivative of Lagrangian (6.106) with respect to b and \mathcal{H} can be found by considering the gradient along the directions of the normal to the bedrock topography and the surface topography respectively [3, Lemma 4] (more details in [109, 110]). In the current optimization problem, the computation is straightforward: Let f be a functional that does not depend on b and \mathcal{H} . Then, one can write

$$\begin{aligned} \frac{\partial}{\partial \mathcal{H}} \left(\int_{\Omega(b, \mathcal{H})} f d\mathbf{x} \right) \hat{\mathcal{H}} &= \frac{\partial}{\partial \mathcal{H}} \left(\int_{\Omega_{\perp}} \int_b^{b+\mathcal{H}} f d\mathbf{x} \right) \hat{\mathcal{H}} = \int_{\Omega_{\perp}} f(x, y, b(x, y) + \mathcal{H}(x, y)) \hat{\mathcal{H}}(x, y) d\mathbf{x} dy \\ \frac{\partial}{\partial b} \left(\int_{\Omega(b, \mathcal{H})} f d\mathbf{x} \right) \hat{b} &= \frac{\partial}{\partial b} \left(\int_{\Omega_{\perp}} \int_b^{b+\mathcal{H}} f d\mathbf{x} \right) \hat{b} = \int_{\Omega_{\perp}} \left(f(x, y, b(x, y) + \mathcal{H}(x, y)) - f(x, y, b(x, y)) \right) \hat{b}(x, y) d\mathbf{x} dy, \end{aligned} \quad (6.107)$$

$$(6.108)$$

for all test functions \hat{b} and $\hat{\mathcal{H}}$.

In the case of a Stokes ice flow, the dual problem reads therefore

$$\begin{cases} -\frac{\partial \lambda}{\partial t} - \mathbf{u} \cdot \nabla \lambda = -2\eta D(\mathbf{u}) : D(\boldsymbol{\psi}) + \mathbf{f} \cdot \boldsymbol{\psi}, & \text{in } \Gamma_s(b, \mathcal{H}) \times [t_i, t_f] \\ \lambda = s^{\text{obs}} - b - \mathcal{H}, & \text{in } \Omega_{\perp} \times \{t_f\} \\ \mu = \lambda, & \text{in } \Omega_{\perp} \times \{t_i\}. \end{cases} \quad (6.109)$$

where the costate variables $\boldsymbol{\psi}$ and χ satisfy

$$\begin{cases} 2\nabla \cdot \left(\tilde{\eta}(\mathbf{u}) D(\mathbf{u}) : D(\boldsymbol{\psi}) D(\mathbf{u}) + \eta(\mathbf{u}) \nabla \boldsymbol{\psi} \right) + \nabla \chi = -\nabla \lambda, & \text{in } \Omega(b, \mathcal{H}) \\ \nabla \cdot \boldsymbol{\psi} = 0, & \text{in } \Omega(b, \mathcal{H}) \\ \boldsymbol{\psi} = 0, & \text{on } \Gamma_b(b, \mathcal{H}) \\ 2\left(\tilde{\eta} D(\mathbf{u}) : D(\boldsymbol{\psi}) D(\mathbf{u}) \mathbf{n} + \eta \frac{\partial \boldsymbol{\psi}}{\partial n} \right) + \chi \mathbf{n} = 0, & \text{on } \Gamma_s(b, \mathcal{H}). \end{cases} \quad (6.110)$$

The dual equations (6.109) of the ice thickness equation (6.104) describe a backward, diffusion-transport problem, which can be reformulated within the VOF formulation [1]. The dual equations (6.110) of the Stokes equations (6.105) correspond to a linear Stokes flow characterized by the Cauchy stress tensor

$$\tilde{\sigma} = 2\left(\tilde{\eta}(\mathbf{u}) D(\mathbf{u}) : D(\boldsymbol{\psi}) D(\mathbf{u}) + \eta \nabla \boldsymbol{\psi} \right) + \chi, \quad (6.111)$$

where

$$\tilde{\eta} = -\frac{2^{\frac{3n-5}{2}} (n-1) \eta^n |D|^{n-3}}{2^{\frac{3n-5}{2}} n |D|^{n-1} \eta^{n-1} + \sigma_0^{n-1}}, \quad (6.112)$$

subjected to the force field

$$\tilde{\mathbf{f}} = -\nabla \lambda. \quad (6.113)$$

The solving of this problem at the continuous level requires the additional formulation of the corresponding KKT conditions [72] which is not developed further here, as it does not bring anything into the sought formal description. Moreover, for the sake of convergent gradients, these equations shall not be discretized, but the procedure proposed in the previous sections for shallow ice flows applied instead, consisting in the differentiation of the discretized equations. Numerically, the additional difficulty in this case is that the linear system changes its shape from one iteration to the other, hence the structure of the computational code has to be changed drastically in order to perform shape optimization.

6.7 Conclusion

The chapter presents a shape optimization algorithm for two and three-dimensional shallow ice flows that estimates a glacier's subglacial topography, or, equivalently, a glacier's ice thickness distribution. The method is based on the minimization of a discrete regularized misfit functional, for the sake of numerical convergence. Indeed, the gradient obtained after differentiation of the forward numerical scheme corresponds to the problem's exact gradient, which is not the case any more when the continuous dual problem and gradient are first deduced from the continuous forward problem by means of calculus of variation and then discretized.

The chapter begins with the formulation of the two-dimensional algorithm that minimizes the surface topography misfit at final time only. A primal–dual method is applied to the simplified, two-dimensional scheme (3.32) for non-sliding glaciers and a time-independent surface mass-balance. Then, its three-dimensional counterpart is

developed that is based on the splitted-in-time numerical scheme (3.14)–(3.18). From its most basic formulation, *i.e.* the direct extension of the two-dimensional algorithm, a complex algorithm is tailored that is capable of incorporating known bedrock topography profiles, surface topographies at several times, and partial surface velocity data. A primal–dual procedure is preferred here over a forward automatic differentiation algorithm for the sake of small computer resources, because the number of control parameters is very high, making a backward sensitivity much more appropriate [57]. A reverse mode automatic differentiation algorithm is too sophisticated for an application to this problem. The method is validated on various synthetical test cases and its sensitivity to the surface data and the model parameters is investigated.

Described by the literature as unreliable in the shallow water framework [43], the shape optimization algorithm applied to the shallow ice equation appears to be accurate, reliable, and fast, as far as the comparisons of Section 6.3 can tell. In particular, even if the initial guess for the bedrock is far from the solution topography, the resulting estimation is close to the solution bedrock on all the tested geometries. The method is as sensitive to surface data as the transient inverse method. As the quasi-stationary and transient inverse methods, the control algorithm needs neither filtering of the surface topography with a lower slope limit nor further assumptions on the model itself. Moreover, it remains reasonably simple as it only consists in the minimization of the misfit between the computed and exact (or measured) surface data and the PDE-constraint is cast into one single evolution equation. The regularization that minimizes the potential and the curvature of the bedrock topography is well-known among glaciologists [63, 85]. A large quantity of minimization algorithms have been efficiently implemented to solve this type of PDE-constrained optimization problems (see *e.g.* [59, 86, 51]).

The SOA can even be used to compute the surface mass-balance parameters and the rate factor in the same time the subglacial bedrock topography is inferred. The numerical experiments presented in Section 6.3.2 show that surface velocity data are really necessary when it comes to the determination of the model parameters other than the bedrock topography. Surface topographies, even if known at several times in the time frame, do not allow for an improved inversion of all the parameters. The parameters of the sliding law defined by (2.8), (2.9), and (2.10) cannot be recovered by any of the methods proposed in this thesis because of their dependence on the sliding altitude z_{sl} that makes the problem ill-posed, as infinitely many solutions exist to the minimization problem, namely that characterized by any sliding altitude and $\mathcal{C} = 0$, or any \mathcal{C} and a sufficiently low sliding altitude such that the gradient of the cost functional with respect to these parameters is zero. In the literature, it has been successfully attempted to compute the subglacial topography together with the sliding coefficient C when it is considered as a function of position and no assumption of the type (2.10) is done on it [46]. However, the method needs a complete grid of surface velocity data that does not exist on Swiss Alpine glaciers.

This chapter is a first attempt of using optimal control techniques for the computation of a glacier's bedrock topography. It demonstrates that the shallow ice equations can be inverted very accurately. The application of the algorithm presented here to real-world data does not supply excellent estimations, because the method is based on the shallow ice approximation. However, perspectives are promising on the inversion of the Stokes problem.

7 Conclusion

This thesis begins by recalling the equations that rule the time evolution of a Stokes ice flow together with its shallow ice approximation. The numerical methods that solve these equations are then presented and the results of various numerical experiments are reported, for the sake of comparison of both ice flow models. On the tested, sliding ice-cap geometries, no significant difference can be noticed between the two approximations. The thinner the glacier's final state, the better the matching of the surfaces. On alpine glaciers, the difference is more important, especially near the tongue of Rhône glacier. A sensitivity analysis of the forward models with respect to the bedrock topography, the rheology, the surface mass-balance, and the sliding law parameters evinces that the inverse problem of inferring the subglacial topography from surface observations is ill-posed. Indeed, many different bedrock topographies lead to equivalent surface topographies of Silvretta glacier after a 10 a-glacier evolution for fixed model parameters, since they differ by less than a finite-element's size everywhere in the computational domain. For example, a 100 m-wide-and-deep bump positioned at several locations in the subglacial topography results most of the time in a surface topography response damped by a factor between 5 and 10.

Nevertheless, several methods are proposed that estimate a glacier's subglacial topography. The quasi-stationary inverse method infers the sought topography from the assumption that the glacier is stationary under the influence of the apparent surface mass-balance. Hence, the evolution's time dependence is hidden in the ice thickness equation's source term and the originally transient, elliptic evolution transforms into a stationary, hyperbolic equation for the ice thickness distribution with Dirichlet boundary conditions. The solving procedure is inspired by an algorithm that reconstructs the bed of steady rivers by means of the steady, shallow water equation [43]. For the sake of numerical stability and convergence, the underlying scheme is regularized.

The transient inverse method relies on the iterative update of the bedrock topography with the surface topography discrepancy. This procedure is inspired from the reconstruction of the velocity field of viscous, gravity-driven flows over an unknown topography [58]. No apparent surface mass-balance is used in this case so that the time evolution is accounted for more accurately. It is developed for both the shallow and Stokes ice flows.

The quasi-stationary and the transient inverse methods belong to the category of gradient-free methods, *i.e.* methods that minimize the surface topography discrepancy without consideration of its gradient with respect to the bedrock topography. They have the advantages of being very easy to implement and relying on very little computer resources at the expense of converging very slowly. Therefore, faster algorithms that make use of gradient informa-

tion are designed in the shallow ice approximation. They consist in the minimization of the surface topography misfit functional that is regularized so as to minimize oscillations in the topography estimation. As the number of control parameters is very large, a primal–dual approach is advocated instead of a forward automatic differentiation method [100]. The regularized misfit functional is therefore augmented to a Lagrangian functional, the stationarity of which yields a simple way to compute the misfit’s gradient with respect to any of the bedrock topography or the model parameters. The three methods are thoroughly compared in the two and three-dimensional shallow ice approximations. Clearly, on the synthetical test cases, the shape optimization algorithm supplies accurate results in little CPU times.

Although the relevance of each of these methods is demonstrated for the inversion of synthetical geometries, their application to real-world measurements does not provide strikingly accurate ice thickness estimations, especially when the transient inverse method is applied to a Stokes ice flow. The best trade-off between computational complexity, computational time, and accuracy is in this case obtained with the three-dimensional, quasi-stationary inverse method. Even though this method relies on the shallow ice approximation, its results are reliable.

The shape optimization method yields promising results in the shallow ice approximation, that gives some zeroth order insight into its extension to Stokes ice flows. Further extensions of the algorithms of this thesis mainly concern the optimal control of a Stokes ice flow, that should supply the best possible algorithm for the estimation of a glacier’s subglacial topography. However, the development of such a shape optimization algorithm still requires a lot of work, as many problems may arise, basically due to the three-dimensional, evolutive, free surface character of the forward problem.

A Linear systems of the forward models

This Appendix details the numerical schemes that allow for the simulation of the evolution in time of Stokes and shallow ice flows. The Stokes ice velocity computed from equation (2.8) is used as the transport velocity in the volume of fluid equation (2.27). The solving of shallow ice flows is simpler because the ice velocity can be expressed analytically. The coming sections first detail the solving of the Stokes equations with a finite-element method. No further details are given on the solving of the VOF equations, as everything is explained thoroughly in [79, 113, 24, 89, 41, 42, 69]. The second section of this Appendix describes how to solve the shallow ice approximation with finite-difference schemes in two and three space dimensions. All the linear systems arising in the next sections are solved with the PETSc Library [7, 8].

A.1 Stokes ice flow

Let us now detail the discretized finite-element vectors and matrices corresponding to the weak formulation (3.9). Formally, the volumic integrals are decomposed into the following sum over every active finite-element K :

$$\int_{\Omega^\ell} (\cdot) = \sum_{\substack{K \in \mathcal{T}_H \\ K \text{ active}}} \int_K (\cdot) \Big|_K \quad (\text{A.1})$$

and the surface integrals into

$$\int_{\Gamma_b^s} (\cdot) = \sum_{\substack{\Delta \in \Gamma_b^s \\ \Delta \text{ active}}} \int_{\Delta} (\cdot) \Big|_{\Delta}. \quad (\text{A.2})$$

Now, the local description of the linear system on element K is detailed. The complete system is then obtained by summing all the contributions over all the active elements. For the sake of readability, on the local element K , we

write

$$\mathbf{u}_H^K = \begin{pmatrix} U_1^1 & U_2^1 & U_3^1 & U_4^1 \\ U_1^2 & U_2^2 & U_3^2 & U_4^2 \\ U_1^3 & U_2^3 & U_3^3 & U_4^3 \end{pmatrix} \begin{pmatrix} \Phi_1^K \\ \Phi_2^K \\ \Phi_3^K \\ \Phi_4^K \end{pmatrix}, \quad \mathbf{v}_H^K = \Phi_k^K \mathbf{e}^\beta, \quad p_H^K = \mathbf{P} \cdot \Phi^K, \quad \text{and} \quad q_H^K = \Phi_k^K, \quad (\text{A.3})$$

where \mathbf{e}^β is the β -th canonical vector of \mathbb{R}^3 and the functions Φ_i^K are the \mathbb{P}_1 -basis functions

$$\Phi_i^K(x, y, z) = \phi_{0i} + \phi_{1i}x + \phi_{2i}y + \phi_{3i}z, \quad 1 \leq i \leq 4, \quad (\text{A.4})$$

for each node of element K . Then, restricted to the local element K , the weak formulation (3.9) is described by the linear system

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & B_1^T \\ A_{21} & A_{22} & A_{23} & B_2^T \\ A_{31} & A_{32} & A_{33} & B_3^T \\ -B_1 & -B_2 & -B_3 & C \end{pmatrix} \begin{pmatrix} \mathbf{U}^1 \\ \mathbf{U}^2 \\ \mathbf{U}^3 \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F}^1 \\ \mathbf{F}^2 \\ \mathbf{F}^3 \\ \mathbf{G} \end{pmatrix}, \quad (\text{A.5})$$

where \mathbf{F}^i and \mathbf{G} correspond to the right-hand sides of (3.9), the vectors \mathbf{U}^j are the vectors of components U_i^j given in (A.3), and the matrices $A_{\beta\gamma}$ and C depend on the ice velocity computed at the previous fixed-point iteration. The components of the matrices B are [29]

$$(B_\beta)_{jk} = - \int_K \Phi_j^K \nabla \cdot (\Phi_k^K \mathbf{e}^\beta) \approx - \frac{\text{vol}(K)}{4} \phi_{\beta k}^K, \quad (\text{A.6})$$

where $\text{vol}(K)$ is the volume of K . Because

$$\int_K \eta^K D(\mathbf{u}_H^K) : D(\mathbf{v}_H^K) = \int_K \eta^K \nabla \mathbf{u}_H^K : \nabla \mathbf{v}_H^K + \sum_{l,j} \int_K \eta^K \partial_l(\mathbf{u}_H^K)_j \partial_j(\mathbf{v}_H^K)_l, \quad (\text{A.7})$$

where η^K is the viscosity evaluated in element K with the velocity determined at the previous fixed-point iteration, we have [29]

$$\begin{aligned} (A_{\beta\gamma})_{jk} &= \int_K \eta^K \sum_{i=1}^3 \frac{\partial \Phi_j^K}{\partial x_i} \frac{\partial \Phi_k^K}{\partial x_i} \delta_{\beta\gamma} + \int_K \eta^K \sum_{i=1}^3 \frac{\partial \Phi_k^K}{\partial x_i} \frac{\partial \Phi_j^K}{\partial x_\beta} \delta_{\gamma i} = \int_K \eta^K \sum_{i=1}^3 \phi_{ij}^K \phi_{ik}^K \delta_{\beta\gamma} + \int_K \eta^K \sum_{i=1}^3 \phi_{ik}^K \phi_{\beta j}^K \delta_{\gamma i} \\ &\approx \text{vol}(K) \eta^K \left(\delta_{\beta\gamma} \sum_{i=1}^3 \left[\phi_{ik}^K \phi_{ij}^K \right] + \phi_{\gamma k}^K \phi_{\beta j}^K \right). \end{aligned} \quad (\text{A.8})$$

Due to sliding, the following expression is added to the matrices A [29]:

$$(A_{\beta\gamma})_{jk} + = \sum_{l=1,2} \int_\Delta \alpha \Phi_\beta^K \Phi_j^K t_{l,k} t_{l,\gamma} - \frac{1}{\varepsilon_{\text{FEM}}} \int_\Delta \Phi_k^K \Phi_j^K n_\beta n_\gamma \approx \frac{\text{vol}(\Delta)}{3} \left(\alpha(\mathbf{N}_\beta) \sum_{l=1,2} t_{l,k}(\mathbf{N}_\beta) t_{l,\gamma}(\mathbf{N}_\beta) \delta_{\beta j} - \frac{\delta_{jk}}{\varepsilon_{\text{FEM}}} n_\beta n_\gamma \right), \quad (\text{A.9})$$

where \mathbf{N}_β is the vector of coordinates of node β of Δ and Δ is the face of K that is contained in Γ_b^S . Due to stabilization, the following corrections must be included [29]:

$$(A_{\beta\gamma})_{jk} + = \int_K \tau_2^K \frac{\partial \Phi_\beta^K}{\partial x_k} \frac{\partial \Phi_j^K}{\partial x_\gamma} \approx \tau_2^K \text{vol}(K) \phi_{k\beta}^K \phi_{\gamma j}^K \quad (\text{A.10})$$

$$C_{jk} = \int_K \tau_1^K \sum_{i=1}^3 \frac{\partial \Phi_k^K}{\partial x_i} \frac{\partial \Phi_j^K}{\partial x_i} \approx \tau_1^K \text{vol}(K) \sum_{i=1}^3 \phi_{ik}^K \phi_{ij}^K, \quad (\text{A.11})$$

where τ_1^K and τ_2^K are τ_1 and τ_2 evaluated in element K with

$$\tau_1^K = \frac{H_K^2}{12\eta^K} \quad \text{and} \quad \tau_2^K = 6\eta^K, \quad (\text{A.12})$$

where H_K is the diameter of element K . Finally, the forcing term reads [29]

$$F_k^\beta = \rho \int_K \mathbf{g} \cdot \mathbf{e}^\beta \Phi_k^K \approx \rho g \delta_{\beta 3} \frac{\text{vol}(K)}{4} \quad (\text{A.13})$$

and the right-hand side, stabilization term is

$$G_k = \rho \int_K \tau_1^K \mathbf{g} \cdot \nabla \Phi_k^K \approx \tau_1^K \rho g \phi_{3k}^K \text{vol}(K). \quad (\text{A.14})$$

The assembly of all these local matrices together in the matrix of the whole triangulation yields to a non-linear system in the velocity and the pressure that is iteratively solved with a fixed-point method. The linearized system is solved with the GMRES method with an ILU preconditioner, available in the PETSc library [7, 8]. Eventually, its solving brings up a velocity that can be given as an input to the VOF Algorithm 2.

A.2 Shallow ice flow

A.2.1 Two-dimensional schemes

Semi-implicit time discretization

Equation (3.33) depends linearly on the components \mathcal{H}_j^ℓ and can be cast into the tridiagonal system of equations

$$\gamma_{j-1}^\ell \mathcal{H}_{j-1}^\ell + \gamma_j^\ell \mathcal{H}_j^\ell + \gamma_{j+1}^\ell \mathcal{H}_{j+1}^\ell = R_j^\ell, \quad (\text{A.15})$$

where

$$\gamma_{j-1}^\ell = -\frac{\Delta t}{\Delta x^2} (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \quad (\text{A.16})$$

$$\gamma_j^\ell = 1 + \frac{\Delta t}{\Delta x^2} \left((\mathcal{D}_f)_{j-\frac{1}{2}}^\ell + (\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \right) \quad (\text{A.17})$$

$$\gamma_{j+1}^\ell = -\frac{\Delta t}{\Delta x^2} (\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \quad (\text{A.18})$$

and

$$R_j^\ell = \mathcal{H}_j^\ell + \frac{\Delta t}{\Delta x} \left((\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \frac{b_{j+1} - b_j}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \frac{b_j - b_{j-1}}{\Delta x} \right). \quad (\text{A.19})$$

Implicit time discretization

Equation (3.35) is non-linear in the ice thickness and is solved with a Newton algorithm. Let us define the non-linear function \mathbf{F} of components

$$F_j(\mathbf{H}) = H_j - \mathcal{H}_j^\ell - \frac{\Delta t}{2\Delta x} \left[(\mathcal{D}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} + (\mathcal{D}_f)_{j+\frac{1}{2}} \frac{b_{j+1} - b_j}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}} \frac{b_j - b_{j-1}}{\Delta x} + (\mathcal{D}_f)_{j+\frac{1}{2}} \frac{H_{j+1} - H_j}{\Delta x} - (\mathcal{D}_f)_{j-\frac{1}{2}} \frac{H_j - H_{j-1}}{\Delta x} \right] - \mathcal{B}_j \Delta t, \quad (\text{A.20})$$

where the $(\mathcal{D}_f)_{j-\frac{1}{2}}$ without time index are evaluated with H_j instead of \mathcal{H}_j . The Jacobian matrix of \mathbf{F} is defined by

$$\begin{aligned} (DF)_{j-1,j} &= \mathcal{C}_{j-\frac{1}{2}}^- \\ (DF)_{j,j} &= 1 + \mathcal{C}_{j-\frac{1}{2}}^+ - \mathcal{C}_{j+\frac{1}{2}}^- \\ (DF)_{j+1,j} &= -\mathcal{C}_{j+\frac{1}{2}}^+, \end{aligned} \quad (\text{A.21})$$

where

$$\mathcal{C}_{j-\frac{1}{2}}^\pm = \frac{\Delta t}{2\Delta x} \left(\frac{n+2}{2} (\mathcal{F}_f)_{j-\frac{1}{2}} \frac{s_j - s_{j-1}}{\Delta x} \pm \frac{n}{\Delta x} (\mathcal{D}_f)_{j-\frac{1}{2}} \right), \quad (\text{A.22})$$

and $(\mathcal{F}_f)_{j-\frac{1}{2}}$ is (6.11) evaluated in H instead of \mathcal{H} .

A.2.2 Three-dimensional schemes

Semi-implicit time discretization

Equations (3.15)-(3.19) depend linearly on the components $\overline{\mathcal{H}}_{jk}^\ell$ and can therefore be cast into the pentadiagonal system of equations

$$\gamma_{j-1,k}^\ell \overline{\mathcal{H}}_{j-1,k}^\ell + \gamma_{j,k-1}^\ell \overline{\mathcal{H}}_{j,k-1}^\ell + \gamma_{jk}^\ell \overline{\mathcal{H}}_{jk}^\ell + \gamma_{j,k+1}^\ell \overline{\mathcal{H}}_{j,k+1}^\ell + \gamma_{j+1,k}^\ell \overline{\mathcal{H}}_{j+1,k}^\ell = R_{jk}^\ell, \quad (\text{A.23})$$

where

$$\gamma_{j-1,k}^\ell = -\frac{\Delta t}{\Delta x^2} \mathcal{D}_{j-\frac{1}{2},k}^\ell \quad (\text{A.24})$$

$$\gamma_{j,k-1}^\ell = -\frac{\Delta t}{\Delta x^2} \mathcal{D}_{j,k-\frac{1}{2}}^\ell \quad (\text{A.25})$$

$$\gamma_{jk}^\ell = 1 + 2\frac{\Delta t}{\Delta x^2} \left(\mathcal{D}_{j,k-\frac{1}{2}}^\ell + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \right) \quad (\text{A.26})$$

$$\gamma_{j,k+1}^\ell = -\frac{\Delta t}{\Delta x^2} \mathcal{D}_{j,k+\frac{1}{2}}^\ell \quad (\text{A.27})$$

$$\gamma_{j+1,k}^\ell = -\frac{\Delta t}{\Delta x^2} \mathcal{D}_{j+\frac{1}{2},k}^\ell \quad (\text{A.28})$$

and

$$R_{jk}^\ell = \mathcal{H}_{jk}^\ell + \frac{\Delta t}{\Delta x} \left(\mathcal{D}_{j+\frac{1}{2},k}^\ell \frac{b_{j+1,k} - b_{jk}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2},k}^\ell \frac{b_{jk} - b_{j-1,k}}{\Delta x} + \mathcal{D}_{j,k+\frac{1}{2}}^\ell \frac{b_{j,k+1} - b_{jk}}{\Delta x} - \mathcal{D}_{j,k-\frac{1}{2}}^\ell \frac{b_{jk} - b_{j,k-1}}{\Delta x} \right) \quad (\text{A.29})$$

Implicit time discretization

Introduce the following quantities:

$$\begin{aligned}
(\mathcal{F}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell &= \Gamma \quad \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n+1,n-1}(\ell) \\
(\mathcal{F}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell &= (\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} \quad \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n,n-1}(\ell) \\
(\mathcal{G}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell &= \Gamma \quad \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n+2,n-3}(\ell) \\
(\mathcal{G}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell &= (\Gamma_s)_{j-\frac{1}{2},k-\frac{1}{2}} \quad \mathcal{K}_{j-\frac{1}{2},k-\frac{1}{2}}^{n+1,n-3}(\ell),
\end{aligned} \tag{A.30}$$

and

$$\mathcal{F}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = (\mathcal{F}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + (\mathcal{F}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell \tag{A.31}$$

$$\mathcal{G}_{j-\frac{1}{2},k-\frac{1}{2}}^\ell = (\mathcal{G}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell + (\mathcal{G}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\ell. \tag{A.32}$$

Equation (3.15) complemented with (3.27) depends non-linearly on the components $\overline{\mathcal{H}}_{jk}^\ell$. It consists in solving

$$\mathbf{F}(\overline{\mathcal{H}}^\ell) = \overline{\mathcal{H}}^\ell - \mathcal{H}^\ell - \Delta t \mathcal{T}(\mathcal{H}^\ell, \overline{\mathcal{H}}^\ell) = 0, \tag{A.33}$$

which is done iteratively with the Newton method. Let \mathbf{w}^ν be the Newton iterate, initialized with $\mathbf{w}^0 = \mathcal{H}^\ell$. Then, solve

$$DF(\mathbf{w}^\nu)(\mathbf{w}^\nu - \mathbf{w}^{\nu+1}) = \mathbf{F}(\mathbf{w}^\nu) \tag{A.34}$$

until convergence, in order to find $\overline{\mathcal{H}}^\ell$, where $DF(\mathbf{w}^\nu)$ is the nonadiagonal Jacobian matrix of elements

$$(DF)_{j-1,k-1;jk} = \frac{\Delta t}{4\Delta x} (\mathcal{C}_2^-)_{jk}^\nu \left(\delta_x \tilde{s}_{jk}^\nu + \delta_y \tilde{s}_{jk}^\nu \right) \tag{A.35}$$

$$(DF)_{j-1,k;jk} = \frac{\Delta t}{4\Delta x} \left((\mathcal{C}_1^-)_{jk}^\nu \left(\delta_x \tilde{s}_{jk}^\nu + \delta_y \tilde{s}_{jk}^\nu \right) - (\mathcal{C}_2^-)_{j,k+1}^\nu \left(\delta_y \tilde{s}_{j,k+1;jk}^\nu - \delta_x \tilde{s}_{jk}^\nu \right) \right) - \frac{\Delta t}{2\Delta x^2} \mathcal{D}_{j-\frac{1}{2},k}^\nu \tag{A.36}$$

$$(DF)_{j-1,k+1;jk} = -\frac{\Delta t}{4\Delta x} (\mathcal{C}_1^-)_{j,k+1}^\nu \left(\delta_y \tilde{s}_{j,k+1}^\nu - \delta_x \tilde{s}_{jk}^\nu \right) \tag{A.37}$$

$$(DF)_{j,k-1;jk} = \frac{\Delta t}{4\Delta x} \left((\mathcal{C}_1^+)_{jk}^\nu \left(\delta_x \tilde{s}_{jk}^\nu + \delta_y \tilde{s}_{jk}^\nu \right) - (\mathcal{C}_2^-)_{j+1,k}^\nu \left(\delta_x \tilde{s}_{j+1,k;jk}^\nu - \delta_y \tilde{s}_{jk}^\nu \right) \right) - \frac{\Delta t}{2\Delta x^2} \mathcal{D}_{j,k-\frac{1}{2}}^\nu \tag{A.38}$$

$$\begin{aligned}
(DF)_{jk;jk} &= 1 - \frac{\Delta t}{4\Delta x} \left[-(\mathcal{C}_2^+)_{jk}^\nu \left(\delta_x \tilde{s}_{jk}^\nu + \delta_y \tilde{s}_{jk}^\nu \right) + (\mathcal{C}_1^-)_{j+1,k}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu - \delta_y \tilde{s}_{jk}^\nu \right) \right. \\
&\quad \left. + (\mathcal{C}_1^+)_{j,k+1}^\nu \left(\delta_y \tilde{s}_{j,k+1}^\nu - \delta_x \tilde{s}_{jk}^\nu \right) + (\mathcal{C}_2^-)_{j+1,k+1}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu + \delta_y \tilde{s}_{j,k+1}^\nu \right) \right] + 2 \frac{\Delta t}{\Delta x^2} \mathcal{D}_{jk}^\nu
\end{aligned} \tag{A.39}$$

$$(DF)_{j,k+1;jk} = -\frac{\Delta t}{4\Delta x} \left((\mathcal{C}_2^+)_{j,k+1}^\nu \left(\delta_y \tilde{s}_{j,k+1}^\nu - \delta_x \tilde{s}_{jk}^\nu \right) + (\mathcal{C}_1^-)_{j+1,k+1}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu + \delta_y \tilde{s}_{j,k+1}^\nu \right) \right) - \frac{\Delta t}{2\Delta x^2} \mathcal{D}_{j,k+\frac{1}{2}}^\nu \tag{A.40}$$

$$(DF)_{j+1,k-1;jk} = -\frac{\Delta t}{4\Delta x} (\mathcal{C}_1^+)_{j+1,k}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu - \delta_y \tilde{s}_{jk}^\nu \right) \tag{A.41}$$

$$(DF)_{j+1,k;jk} = -\frac{\Delta t}{4\Delta x} \left((\mathcal{C}_2^+)_{j+1,k}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu - \delta_y \tilde{s}_{jk}^\nu \right) + (\mathcal{C}_1^+)_{j+1,k+1}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu + \delta_y \tilde{s}_{j,k+1}^\nu \right) \right) - \frac{\Delta t}{2\Delta x^2} \mathcal{D}_{j+\frac{1}{2},k}^\nu \tag{A.42}$$

$$(DF)_{j+1,k+1;jk} = -\frac{\Delta t}{4\Delta x} (\mathcal{C}_2^+)_{j+1,k+1}^\nu \left(\delta_x \tilde{s}_{j+1,k}^\nu + \delta_y \tilde{s}_{j,k+1}^\nu \right), \tag{A.43}$$

Appendix A. Linear systems of the forward models

where $\mathcal{D}_{j-\frac{1}{2},k-\frac{1}{2}}^\nu$ is the discretization (3.20) of (2.20)-(2.21) evaluated in w_{jk}^ν instead of \mathcal{H}_{jk}^ℓ ,

$$\tilde{s}_{jk}^\nu = b_{jk} + w_{jk}^\nu, \quad \delta_x \tilde{s}_{jk} = \frac{\tilde{s}_{jk} - \tilde{s}_{j-1,k}}{\Delta x}, \quad \delta_y \tilde{s}_{jk} = \frac{\tilde{s}_{jk} - \tilde{s}_{j,k-1}}{\Delta x}, \quad (\text{A.44})$$

and

$$(\mathcal{C}_1^\pm)_{jk}^\nu = \frac{n+2}{4}(\mathcal{F}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu + \frac{n+1}{4}(\mathcal{F}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu \pm \frac{n-1}{2\Delta x^2} \mathcal{G}_{j-\frac{1}{2},k-\frac{1}{2}}^\nu \left(\tilde{s}_{j,k-1}^\nu - \tilde{s}_{j-1,k}^\nu \right) \quad (\text{A.45})$$

$$(\mathcal{C}_2^\pm)_{jk}^\nu = \frac{n+2}{4}(\mathcal{F}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu + \frac{n+1}{4}(\mathcal{F}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu \pm \frac{n-1}{2\Delta x^2} \mathcal{G}_{j-\frac{1}{2},k-\frac{1}{2}}^\nu \left(\tilde{s}_{jk}^\nu - \tilde{s}_{j-1,k-1}^\nu \right), \quad (\text{A.46})$$

where $(\mathcal{F}_s)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu$, $(\mathcal{F}_f)_{j-\frac{1}{2},k-\frac{1}{2}}^\nu$, and $\mathcal{G}_{j-\frac{1}{2},k-\frac{1}{2}}^\nu$ are (A.30) and (A.32), evaluated in w_{jk}^ν instead of \mathcal{H}_{jk}^ℓ .

B Linear systems of the dual problems

This Appendix deals with the linear systems arising during the solving of the dual problems in the optimal control algorithms of Chapter 6. The linear systems appearing in this Appendix are solved with the PETSc Library [7, 8]. The line search as well as the BFGS algorithm (*e.g.* [15, 35, 47, 107] and [88, Chapter 6.1]) are realized with the Toolkit for Advanced Optimization (TAO) method “Bounded Limited Memory Variable Metric”, `-tao_blmvm`, with the default values for its parameters [86], save for its convergence parameters. The choice of the bounded version of LMVM is due to bounds (6.24) and (6.25).

B.1 Two-dimensional formulation

B.1.1 Semi-implicit time discretization

Problem (6.7) with right-hand side (6.10) is linear in λ^ℓ and is cast into the tridiagonal linear system of equations

$$\gamma_{j-1}^{\ell-1} \lambda_{j-1}^\ell + \gamma_j^{\ell-1} \lambda_j^\ell + \gamma_{j+1}^{\ell-1} \lambda_{j+1}^\ell = R_j^\ell, \quad (\text{B.1})$$

where the γ_j^ℓ are defined by (A.18) and

$$\begin{aligned} R_j^\ell = \lambda_j^{\ell+1} - \Delta t \left[\left(\frac{n+2}{2} (\mathcal{F}_f)_{j+\frac{1}{2}}^\ell - \frac{n-1}{\Delta x} (\mathcal{G}_f)_{j+\frac{1}{2}}^\ell \right) \frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} \right. \\ \left. + \left(\frac{n+2}{2} (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell + \frac{n-1}{\Delta x} (\mathcal{G}_f)_{j-\frac{1}{2}}^\ell \right) \frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x} \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right]. \end{aligned} \quad (\text{B.2})$$

B.1.2 Implicit time discretization

Problem (6.7) with right-hand side (6.15) is also linear in λ^ℓ and cast into the tridiagonal linear system of equations (B.1), where

$$\gamma_{j-1}^\ell = -\frac{\Delta t}{2\Delta x} \left(\frac{n+2}{2} (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} + \frac{n}{\Delta x} (\mathcal{D}_s)_{j-\frac{1}{2}}^\ell \right) \quad (\text{B.3})$$

$$\gamma_j^\ell = 1 + \frac{\Delta t}{2\Delta x} \left(\frac{n+2}{2} (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} + \frac{n}{\Delta x} (\mathcal{D}_s)_{j-\frac{1}{2}}^\ell - \frac{n+2}{2} (\mathcal{F}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} + \frac{n}{\Delta x} (\mathcal{D}_s)_{j+\frac{1}{2}}^\ell \right) \quad (\text{B.4})$$

$$\gamma_{j+1}^\ell = \frac{\Delta t}{2\Delta x} \left(\frac{n+2}{2} (\mathcal{F}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} - \frac{n}{\Delta x} (\mathcal{D}_s)_{j+\frac{1}{2}}^\ell \right) \quad (\text{B.5})$$

and

$$R_j^\ell = \lambda_j^{\ell+1} - \frac{\Delta t}{2} \left[\left(\frac{n+2}{2} (\mathcal{F}_f)_{j+\frac{1}{2}}^\ell \frac{s_{j+1}^\ell - s_j^\ell}{\Delta x} - \frac{n}{\Delta x} (\mathcal{D}_s)_{j+\frac{1}{2}}^\ell \right) \frac{\lambda_{j+1}^{\ell+1} - \lambda_j^{\ell+1}}{\Delta x} - \left(\frac{n+2}{2} (\mathcal{F}_f)_{j-\frac{1}{2}}^\ell \frac{s_j^\ell - s_{j-1}^\ell}{\Delta x} + \frac{n}{\Delta x} (\mathcal{D}_s)_{j-\frac{1}{2}}^\ell \right) \frac{\lambda_j^{\ell+1} - \lambda_{j-1}^{\ell+1}}{\Delta x} \right]. \quad (\text{B.6})$$

B.2 Three-dimensional formulation

B.2.1 Semi-implicit time discretization

Equation (6.40) with right-hand side (6.55) is linear in $\bar{\lambda}^\ell$ and can be cast into the pentadiagonal system of equations

$$\gamma_{j-1,k}^\ell \bar{\lambda}_{j-1,k}^\ell + \gamma_{j,k-1}^\ell \bar{\lambda}_{j,k-1}^\ell + \gamma_{j,k}^\ell \bar{\lambda}_{j,k}^\ell + \gamma_{j,k+1}^\ell \bar{\lambda}_{j,k+1}^\ell + \gamma_{j+1,k}^\ell \bar{\lambda}_{j+1,k}^\ell = \bar{\lambda}_{j,k}^\ell, \quad (\text{B.7})$$

where the coefficients γ are defined by (A.28).

B.2.2 Implicit time discretization

Equation (6.40) with right-hand side (6.60) is linear in $\bar{\lambda}^\ell$, *i.e.* it can be expressed as the nonadiagonal system

$$\gamma_{j-1,k-1}^\ell \bar{\lambda}_{j-1,k-1}^\ell + \gamma_{j,k-1}^\ell \bar{\lambda}_{j,k-1}^\ell + \gamma_{j+1,k-1}^\ell \bar{\lambda}_{j+1,k-1}^\ell + \gamma_{j-1,k}^\ell \bar{\lambda}_{j-1,k}^\ell + \gamma_{j,k}^\ell \bar{\lambda}_{j,k}^\ell + \gamma_{j+1,k}^\ell \bar{\lambda}_{j+1,k}^\ell + \gamma_{j-1,k+1}^\ell \bar{\lambda}_{j-1,k+1}^\ell + \gamma_{j,k+1}^\ell \bar{\lambda}_{j,k+1}^\ell + \gamma_{j+1,k+1}^\ell \bar{\lambda}_{j+1,k+1}^\ell = \bar{\lambda}_{j,k}^\ell, \quad (\text{B.8})$$

where the coefficients γ are defined by

$$\gamma_{j-1,k-1}^\ell = -\frac{\Delta t}{4\Delta x} (\overline{\mathcal{C}_2^+})_{jk}^\ell \left(\nabla_x \bar{s}_{j,k-1}^\ell + \nabla_y \bar{s}_{j-1,k}^\ell \right) \quad (\text{B.9})$$

$$\gamma_{j,k-1}^\ell = -\frac{\Delta t}{2\Delta x^2} \overline{\mathcal{D}}_{j,k-\frac{1}{2}}^\ell + \frac{\Delta t}{4\Delta x} \left[(\overline{\mathcal{C}_2^+})_{jk}^\ell \left(\nabla_x \bar{s}_{j,k-1}^\ell - \nabla_y \bar{s}_{jk}^\ell \right) - (\overline{\mathcal{C}_1^-})_{j+1,k}^\ell \left(\nabla_x \bar{s}_{j+1,k-1}^\ell + \nabla_y \bar{s}_{jk}^\ell \right) \right] \quad (\text{B.10})$$

$$\gamma_{j+1,k-1}^\ell = \frac{\Delta t}{4\Delta x} (\overline{\mathcal{C}_1^-})_{j+1,k}^\ell \left(\nabla_x \bar{s}_{j+1,k-1}^\ell - \nabla_y \bar{s}_{j+1,k}^\ell \right) \quad (\text{B.11})$$

$$\gamma_{j-1,k}^\ell = -\frac{\Delta t}{2\Delta x^2} \overline{\mathcal{D}}_{j-\frac{1}{2},k}^\ell + \frac{\Delta t}{4\Delta x} \left[(\overline{\mathcal{C}_2^+})_{jk}^\ell \left(\nabla_y \bar{s}_{j-1,k}^\ell - \nabla_x \bar{s}_{jk}^\ell \right) - (\overline{\mathcal{C}_1^+})_{j,k+1}^\ell \left(\nabla_x \bar{s}_{jk}^\ell + \nabla_y \bar{s}_{j-1,k+1}^\ell \right) \right] \quad (\text{B.12})$$

$$\gamma_{jk}^\ell = 1 + 2 \frac{\Delta t}{\Delta x^2} \overline{\mathcal{D}}_{jk}^\ell + \frac{\Delta t}{4\Delta x} \left[(\overline{\mathcal{C}_1^-})_{j+1,k}^\ell \left(\nabla_y \bar{s}_{jk}^\ell - \nabla_x \bar{s}_{j+1,k}^\ell \right) + (\overline{\mathcal{C}_2^+})_{jk}^\ell \left(\nabla_x \bar{s}_{jk}^\ell + \nabla_y \bar{s}_{jk}^\ell \right) \right] \quad (\text{B.13})$$

$$- (\overline{\mathcal{C}_2^-})_{j+1,k+1}^\ell \left(\nabla_x \bar{s}_{j+1,k}^\ell + \nabla_y \bar{s}_{j,k+1}^\ell \right) + (\overline{\mathcal{C}_1^+})_{j,k+1}^\ell \left(\nabla_x \bar{s}_{jk}^\ell - \nabla_y \bar{s}_{j,k+1}^\ell \right) \right] \quad (\text{B.14})$$

$$\gamma_{j+1,k}^\ell = -\frac{\Delta t}{2\Delta x^2} \overline{\mathcal{D}}_{j+\frac{1}{2},k}^\ell + \frac{\Delta t}{4\Delta x} \left[(\overline{\mathcal{C}_1^-})_{j+1,k}^\ell \left(\nabla_x \bar{s}_{j+1,k}^\ell + \nabla_y \bar{s}_{j+1,k}^\ell \right) + (\overline{\mathcal{C}_2^-})_{j+1,k+1}^\ell \left(\nabla_x \bar{s}_{j+1,k}^\ell - \nabla_y \bar{s}_{j+1,k+1}^\ell \right) \right] \quad (\text{B.15})$$

$$\gamma_{j-1,k+1}^\ell = \frac{\Delta t}{4\Delta x} \overline{(\mathcal{C}_1^+)}^\ell_{j,k+1} \left(\nabla_y \bar{s}_{j-1,k+1}^\ell - \nabla_x \bar{s}_{j,k+1}^\ell \right) \quad (\text{B.16})$$

$$\gamma_{j,k+1}^\ell = -\frac{\Delta t}{2\Delta x^2} \overline{\mathcal{D}}^\ell_{j,k+\frac{1}{2}} + \frac{\Delta t}{4\Delta x} \left[\overline{(\mathcal{C}_2^-)}^\ell_{j+1,k+1} \left(\nabla_y \bar{s}_{j,k+1}^\ell - \nabla_x \bar{s}_{j+1,k+1}^\ell \right) + \overline{(\mathcal{C}_1^+)}^\ell_{j,k+1} \left(\nabla_x \bar{s}_{j,k+1}^\ell + \nabla_y \bar{s}_{j,k+1}^\ell \right) \right] \quad (\text{B.17})$$

$$\gamma_{j+1,k+1}^\ell = \frac{\Delta t}{4\Delta x} \overline{(\mathcal{C}_2^-)}^\ell_{j+1,k+1} \left(\nabla_x \bar{s}_{j+1,k+1}^\ell + \nabla_y \bar{s}_{j+1,k+1}^\ell \right), \quad (\text{B.18})$$

where $\overline{(\mathcal{C}_1^+)}$ and $\overline{(\mathcal{C}_2^\pm)}$ are defined by (A.45) and (A.46) respectively save that they are evaluated in $\overline{\mathcal{H}}_{jk}^\ell$ instead of w_{jk}^ν .

Bibliography

- [1] K. Abe and K. Koro. A topology optimization approach using vof method. *Structural and Multidisciplinary Optimization*, 31:470–479, 2006. 10.1007/s00158-005-0582-5.
- [2] G. Adalgeirsdottir. Flow dynamics of Vatnajökull ice cap, Iceland, 2003. Mitteilung 181, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie der ETH Zürich.
- [3] G. Allaire, F. Jouve, and A. M. Toader. Structural optimization using sensitivity analysis and a level-set method. *JOURNAL OF COMPUTATIONAL PHYSICS*, 194:363–393, FEB 10 2004.
- [4] R. J. Arthern and G. H. Gudmundsson. Initialization of ice-sheet forecasts viewed as an inverse Robin problem. *Journal of Glaciology*, 56:527–533, 2010.
- [5] S. Avdonin, V. Kozlov, D. Maxwell, and M. Truffer. Iterative methods for solving a nonlinear boundary inverse problem in glaciology. *Journal of inverse and ill-posed problems*, 17:239–258, APR 2009.
- [6] D. B. Bahr, M. F. Meier, and S. D. Peckham. The physical basis of glacier volume-area scaling. *Journal of Geophysical Research*, 102:20355–20362, 1997.
- [7] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical report, Argonne National Laboratory, 2012. ANL-95/11 - Revision 3.3.
- [8] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2011. <http://www.mcs.anl.gov/petsc>.
- [9] R. Becker, H. Kapp, and R. Rannacher. Adaptive Finite Element Methods for Optimal Control of Partial Differential Equations: Basic Concept. *SIAM Journal on Control and Optimization*, 39(1):113–132, 2000.
- [10] H. Blatter. Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients. *J. Glaciol.*, 41(138):333–344, 1995.
- [11] Andrea Bonito, Marco Picasso, and Manuel Laso. Numerical simulation of 3d viscoelastic flows with free surfaces. *Journal of Computational Physics*, 215(2):691 – 716, 2006.
- [12] J. Bourgain and H. Brezis. On the equation $\operatorname{div} y = f$ and application to control of phases. *Journal of the American Mathematical Society*, 16(2):393–426, 2003.
- [13] P. Bousquet, P. Mironescu, and E. Russ. A limiting case for the divergence equation, Nov 2011. Unpublished, available on Hyper Articles en Ligne <http://hal.archives-ouvertes.fr/hal-00639326>.
- [14] I. C. Briggs. Machine contouring using minimum curvature. *Geophysics*, 39(1):39–48, 1974.
- [15] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *Journal of Applied Mathematics*, 6(1):76–90, Mar. 1970.

- [16] W. F. Budd, R. C. Warner, T. H. Jacka, J. Li, and A. Treverrow. Ice flow relations for stress and strain-rate components from combined shear and compression laboratory experiments. *Journal of Glaciology*, 59(214):374–392, 2013.
- [17] E. Bueler. Numerical approximation of a two-dimensional thermomechanical model for ice flow. Department of mathematical sciences, technical report 02-02, University of Alaska, Fairbanks, 2002.
- [18] E. Bueler, C. S. Lingle, J. A. Kallen-Brown, D. N. Covey, and L. N. Bowman. Exact solutions and verification of numerical models for isothermal ice sheets. *Journal of Glaciology*, 51(173):291–306, 2005.
- [19] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.
- [20] P. Burlando, F. Pellicciotti, U. Strasser, L. N. Braun, E. Brun, and R. Kirnbauer. Modelling mountainous water systems between learning and speculating looking for challenges. *Nordic hydrology*, 33(1):47–74, 2002.
- [21] A. Caboussat, M. Picasso, and J. Rappaz. Numerical simulation of free surface incompressible liquid flows surrounded by compressible gas. *Journal of Computational Physics*, 203(2):626 – 649, 2005.
- [22] H. Chen, C. Miao, and X. Lv. A three-dimensional numerical internal tidal model involving adjoint method. *International Journal for Numerical Methods in Fluids*, 69(10):1584–1613, 2012.
- [23] J. Chen and A. Ohmura. *Estimation of Alpine glacier water resources and their change since the 1870s*. International Association of Hydrological Sciences, 1990. Proceedings of two Lausanne Symposia, No. 193.
- [24] A. J. Chorin. Flame advection and propagation algorithms. *Journal of Computational Physics*, 35(1):1–11, 1980.
- [25] E. L. Christensen, N. Reeh, R. Forsberg, J. H. Jorgensen, N. Skou, and K. Woelders. A low-cost glacier-mapping system. *Journal of Glaciology*, 46(154):531–537, 2000.
- [26] G. K. C. Clarke, F. S. Anslow, A. H. Jarosch, V. Radic, B. Menounos, T. Bolch, and E. Berthier. Ice volume and subglacial topography for western canadian glaciers from mass balance fields, thinning rates, and a bed stress model. *Journal of Climate*, 2012. at press.
- [27] G. K. C. Clarke, E. Berthier, C. Schoof, and A. H. Jarosch. Neural Networks Applied to Estimating Subglacial Topography and Glacier Volume. *Journal of Climate*, 22:2146–2160, APR 2009.
- [28] Jacques Colinge and Jacques Rappaz. A strongly nonlinear problem arising in glaciology. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(02):395–406, 1999.
- [29] C. Cuvelier, A. Segal, and A. A. van Steenhoven. *Finite Element Methods and Navier-Stokes Equations*. D. Reidel Publishing Company, 1st edition, 1986.
- [30] R. Delaloye, A. Turatti, W. Haeberli, and H. Holzhauser. *Operation Glaciers*.
- [31] Jean Donea. Recent advances in computational methods for steady and transient transport problems. *Nuclear Engineering and Design*, 80(2):141 – 162, 1984. 4th Special Issue on Smirt-7.
- [32] M. Zemp et al. *Global Outlook for Ice and Snow*. United Nations Environment Programme Nairobi, 2007. pp. 115–152.
- [33] D. Farinotti, H. Corr, and G. H. Gudmundsson. The ice thickness distribution of Flask Glacier, Antarctic Peninsula, determined by combining radio-echo soundings, surface velocity data, and flow modelling. *Annals of Glaciology*, 54(63):18–24, 2012.
- [34] D. Farinotti, M. Huss, A. Bauder, M. Funk, and M. Truffer. A method to estimate the ice volume and ice-thickness distribution of alpine glaciers. *Journal of Glaciology*, 55(191):422–430, 2009.
- [35] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.
- [36] A. C. Fowler. A theoretical treatment of the sliding of glaciers in the absence of cavitation. *Philos. Trans. Roy. Soc. London Ser. A*, 298(1445):637–681, 1980/81.

- [37] L. P. Franca and S. L. Frey. Stabilized finite element methods: II. the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99:209 – 233, 1992.
- [38] Leopoldo P. Franca and Thomas J.R. Hughes. Convergence analyses of galerkin least-squares methods for symmetric advective-diffusive forms of the stokes and incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 105(2):285 – 298, 1993.
- [39] O. Gagliardini, D. Cohen, P. Råback, and T. Zwinger. Finite-element modeling of subglacial cavities and related friction law. *J. Geophys. Res.*, 112, 2007.
- [40] F. Gallaire. Adjoint methods for flow control. *European Research Community on Flow, Turbulence and Combustion Bulletin*, (73):27–32, 2007.
- [41] C. A. Gandin, J. L. Desbiolles, M. Rappaz, and P. Thevoz. A three-dimensional cellular automaton-finite element model for the prediction of solidification grain structures. *Metallurgical and Materials Transactions A-Physical Metallurgy and Material Science*, 30(12):3153–3165, DEC 1999.
- [42] Ch.-A Gandin and M Rappaz. A 3d cellular automaton algorithm for the prediction of dendritic grain growth. *Acta Materialia*, 45(5):2187 – 2195, 1997.
- [43] A. F. Gessese, M. Sellier, E. Van Houten, and G. Smart. Reconstruction of river bed topography from free surface data using a direct numerical approach in one-dimensional shallow water flow. *Inverse Problems*, 27(2):025001, 2011.
- [44] J. W. Glen. The flow law of ice. *IUGG/IAHS Symposium of Chamonix IAHS Publication*, 47:171–183, 1958.
- [45] Roland Glowinski. Finite element methods for incompressible viscous flow. In P.G. Ciarlet and J.L. Lions, editors, *Numerical Methods for Fluids (Part 3)*, volume 9 of *Handbook of Numerical Analysis*, pages 3 – 1176. Elsevier, 2003.
- [46] D. N. Goldberg and O. V. Sergienko. Data assimilation using a hybrid ice flow model. *The Cryosphere*, 5(2):315–327, 2011.
- [47] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.
- [48] R. Greve and H. Blatter. *Dynamics of Ice Sheets and Glaciers*. Springer, 1st edition, 2009.
- [49] G. H. Gudmundsson. A three-dimensional numerical model of the confluence area of unteraargletscher, bernese alps, switzerland. *Journal of Glaciology*, 45(150):219–230, 1999.
- [50] G. H. Gudmundsson, T. Thorsteinsson, and C. Raymond. Inferring bed topography and stickiness from surface data on ice streams. *American Geophysical Union Fall Meeting Abstracts*, dec 2001. Abstract number IP21A-0678.
- [51] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2013.
- [52] A. Guégan, P. J. Schmid, and P. Huerre. Optimal energy growth and optimal control in swept hiemenz flow. *Journal of Fluid Mechanics*, 566:11–45, 10 2006.
- [53] M. Habermann, D. Maxwell, and M. Truffer. Reconstruction of basal properties in ice sheets using iterative inverse methods. *Journal of Glaciology*, 58(210):795–807, 2012.
- [54] W. Haeberli and M. Hoelzle. Application of inventory data for estimating characteristics of and regional climate-change effects on mountain glaciers: a pilot study with the European Alps. *Annals of Glaciology*, 21:206–212, 1995.
- [55] P. Halfar. On the dynamics of the ice sheets. *Journal of Geophysical Research: Oceans*, 86:11065–11072, 1981.
- [56] P. Halfar. On the dynamics of the ice sheets 2. *Journal of Geophysical Research: Oceans*, 88:6043–6051, 1983.
- [57] P. Heimbach and V. Bugnion. Greenland ice-sheet volume sensitivity to basal, surface and initial conditions derived from an adjoint model. *Annals of Glaciology*, 50(52):67–80, 2009.

- [58] C. Heining. Velocity field reconstruction in gravity-driven flow over unknown topography. *Physics of Fluids*, 23(3):032101, Mar. 2011.
- [59] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.
- [60] R. LeB. Hooke. Principles of glacier mechanics. *Cambridge*, 2005.
- [61] M. Huss, A. Bauder, M. Funk, and R. Hock. Determination of the seasonal mass balance of four Alpine glaciers since 1865. *Journal of Geophysical Research*, 113:F01015, 2008.
- [62] M. Huss, R. Hock, A. Bauder, and M. Funk. 100-year mass changes in the swiss alps linked to the atlantic multidecadal oscillation. *Geophys. Res. Lett.*, 37, 2010.
- [63] M. F. Hutchinson. A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. *Journal of Hydrology*, 106:211–232, 1989.
- [64] M. F. Hutchinson, J. A. Stein, J. L. Stein, and T. Xu. Locally adaptive gridding of noisy high resolution topographic data. *18th World IMACS / MODSIM Congress*, July 2009.
- [65] K. Hutter. *Theoretical Glaciology*. Reidel, 1983.
- [66] G. Juvet. Modélisation, analyse mathématique et simulation numérique de la dynamique des glaciers, 2010. EPFL PhD thesis no. 4677.
- [67] G. Juvet, M. Huss, H. Blatter, M. Picasso, and J. Rappaz. Numerical simulation of Rhonegletscher from 1874 to 2100. *J. Comp. Phys.*, 228:6426–6439, 2009.
- [68] G. Juvet, M. Huss, M. Funk, and H. Blatter. Modelling the retreat of grosser aletschgletscher, switzerland, in a changing climate. *Journal of Glaciology*, 57(206):1033–1045, 2011.
- [69] G. Juvet, M. Picasso, J. Rappaz, and H. Blatter. A new algorithm to simulate the dynamics of a glacier: theory and applications. *Journal of Glaciology*, 54(188):801–811, 2008.
- [70] A. Khazendar, E. Rignot, and E. Larour. Larsen B Ice Shelf rheology preceding its disintegration inferred by a control method. *Geophysical Research Letters*, 34, OCT 4 2007.
- [71] P. Kirner. *Modélisation mathématique et simulation numérique des phénomènes dynamiques et thermiques apparaissant dans un glacier*. PhD thesis, EPFL, 2007.
- [72] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.
- [73] D. Kuzmin. A guide to numerical methods for transport equations. *Free CFD Books*, 2010. University of Nürnberg.
- [74] E. Larour, E. Rignot, I. Joughin, and D. Aubry. Rheology of the Ronne Ice Shelf, Antarctica, inferred from satellite radar interferometry data using an inverse control method. *Geophysical Research Letters*, 32(5), MAR 8 2005.
- [75] H. L. Li, F. Ng, Z. Q. Li, D. H. Qin, and G. D. Cheng. An extended "perfect-plasticity" method for estimating ice thickness along the flow line of mountain glaciers. *Journal of Geophysical Research*, 117, 2012. F01020.
- [76] A. Linsbauer, F. Paul, and W. Haeberli. Modeling glacier thickness distribution and bed topography over entire mountain ranges with GlabTop: Application of a fast and robust approach. *Journal of Geophysical Research*, 117, 2012. F03007.
- [77] S. Lovie. Interpreting multivariate data - Proceedings of the conference entitled looking at multivariate data held in Sheffield, UK, March 1980 - BARNETT,V. *British Journal of Mathematical & Statistical Psychology*, 35:234–235, 1982.

- [78] M. W. Mahaffy. 3-dimensional numerical model of ice sheets - tests on bernes ice cap. *Journal of geophysical research - Oceans and Atmospheres*, 81(6):1059–1066, 1976.
- [79] V. Maronnier. *Simulation numérique d'écoulements de fluides incompressibles avec surface libre*. PhD thesis, EPFL, 2000.
- [80] V. Maronnier, M. Picasso, and J. Rappaz. Numerical simulation of three-dimensional free surface flows. *International Journal for Numerical Methods in Fluids*, 42(7):697–716, 2003.
- [81] N. Martin. *Modélisations directes et inverses d'écoulements géophysiques par méthodes variationnelles - Application à la glaciologie*. PhD thesis, Université de Toulouse, 2013.
- [82] D. Maxwell, M. Truffer, S. Avdonin, and M. Stuefer. An iterative scheme for determining glacier velocities and stresses. *Journal of Glaciology*, 54:888–898, 2008.
- [83] R. W. McNabb, R. Hock, S. O'Neel, L. A. Rasmussen, Y. Ahn, M. Braun, H. B. Conway, S. Herreid, I. R. Joughin, W. T. Pfeffer, B. E. Smith, and M. Truffer. Using surface velocities to calculate ice thickness and bed topography: a case study at Columbia Glacier, Alaska, USA. *Journal of Glaciology*, 58(212):1151–1164, 2012.
- [84] W. Mittelholzer. *Les Ailes et les Alpes. Ouvrage publié avec la collaboration de H. Kempf, Berne, vétéran C.A.S. Adaptation de René Gouzy*. impression Paul Attinger (S.A.), 1928.
- [85] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry. A mass conservation approach for mapping glacier ice thickness. *Geophysical Research Letters*, 38:L19503, 2011.
- [86] T. Munson, J. Sarich, S. Wild, S. Benson, and L. C. McInnes. Tao 2.0 users manual. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, 2012. <http://www.mcs.anl.gov/tao>.
- [87] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971. 10.1007/BF02995904.
- [88] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006.
- [89] W.F. Noh and Paul Woodward. Slic (simple line interface calculation). In AdriaanI. Vooren and PieterJ. Zandbergen, editors, *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede*, volume 59 of *Lecture Notes in Physics*, pages 330–340. Springer Berlin Heidelberg, 1976.
- [90] J. F. Nye. A comparison between the theoretical and the measured long profile of the Unteraar Glacier. *Journal of Glaciology*, 2(12):103–107, 1952.
- [91] J. F. Nye. The mechanics of glacier flow. *Journal of Glaciology*, 2(12):82–93, 1952.
- [92] W. S. B. Paterson and K. M. Cuffey. *The physics of glaciers*. Pergamon, Tarrytown, 4th edition, 2010.
- [93] F. Paul and A. Linsbauer. Modeling of glacier bed topography from glacier outlines, central branch lines, and a DEM. *International Journal of Geographical Information Science*, 26(7):1173–1190, 2012.
- [94] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas. An inexact gauss newton method for inversion of basal sliding and rheology parameters in a nonlinear stokes ice sheet model. *Journal of Glaciology*, 58(211):889–903, 2012.
- [95] M. Picasso, J. Rappaz, A. Reist, M. Funk, and H. Blatter. Numerical simulation of the motion of a two dimensional glacier, 2000.
- [96] V. Radic and R. Hock. Regionally differentiated contribution of mountain glaciers and ice caps to future sea-level rise. *Nature Geoscience*, 4:91–94, FEB 2011.
- [97] J. Rappaz and A. Reist. Mathematical and numerical analysis of a three-dimensional fluid flow model in glaciology. *Mathematical models & methods in applied sciences*, 15:37–52, JAN 2005.

- [98] M. J. Raymond and G. H. Gudmundsson. Estimating basal properties of glaciers from surface measurements: a non-linear Bayesian inversion approach. *The Cryosphere Discussions*, 3(1):181–222, 2009.
- [99] M. Raymond-Pralong and G. H. Gudmundsson. Bayesian estimation of basal conditions on rutford ice stream, west antarctica, from surface data. *Journal of Glaciology*, 57(202):315–324, 2011.
- [100] R. Roth and S. Ulbrich. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow, Turbulence and Combustion*, 90(4):763–783, 2013.
- [101] C. Rothenbuehler. Erfassung und darstellung der geomorphologie im gebiet bernina (gr) mit hilfe von gis. *Physische Geographie*, 41:117–126, 2003.
- [102] R Scardovelli and S Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.
- [103] M. Schafer and E. Le Meur. Improvement of a 2-d sia ice-flow model: application to glacier de saint-sorlin, france. *Journal of Glaciology*, 53(183):713–722, 2007.
- [104] C. Schoof. The effect of cavitation on glacier sliding. *Royal Society of London Proceedings Series A*, 461:609–627, March 2005.
- [105] C. Schoof. Coulomb friction and other sliding laws in a higher-order glacier flow model. *M3AS*, submitted, 2009.
- [106] C. Schoof. Ice-sheet acceleration driven by melt supply variability. *Nature*, 468:803–806, Dec. 2010.
- [107] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.
- [108] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd Association for Computing Machinery National Conference*, pages 517–524, New York, NY, USA, Feb. 1968. Association for Computing Machinery.
- [109] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numerical Functional Analysis and Optimization*, 2(7):649–687, 1980.
- [110] J. Sokolowski and J.-P. Zolesio. *Introduction to Shape Optimization. Shape Sensitivity Analysis*. Springer Series in Computational Mathematics. Springer, 1992.
- [111] T. Thorsteinsson, C. F. Raymond, G. H. Gudmundsson, R. A. Bindshadler, P. Vornberger, and I. Joughin. Bed topography and lubrication inferred from surface measurements on fast-flowing ice streams. *Journal of Glaciology*, 49(167):481–490, 2003.
- [112] A. N. Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.
- [113] L. Tobiska. Pironneau, o., finite element methods for fluids, chichester etc., john wiley & sons, paris etc., masson 1989. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 72(7):302–302, 1992.
- [114] W. J. J. van Pelt, J. Oerlemans, C. H. Reijmer, R. Pettersson, V. A. Pohjola, E. Isaksson, and D. Divine. An iterative inverse method to estimate basal topography and initialize ice flow models. *The Cryosphere*, 7(3):987–1006, 2013.
- [115] A. Vieli, M. Funk, and H. Blatter. Tidewater glaciers: frontal flow acceleration and basal sliding. *Annals of Glaciology*, 31:217–221(5), January 2000.
- [116] A. Vieli and A. J. Payne. Application of control methods for modelling the flow of pine island glacier, west antarctica. *Annals of Glaciology*, 36(1):197–204, 2003.
- [117] J. Weertman. On the sliding of glaciers. *Journal of Glaciology*, 3(21):33–38, 1957.

- [118] N.K. Yamaleev, B. Diskin, and E.J. Nielsen. Adjoint-based methodology for time-dependent optimization. *12th The American Institute of Aeronautics and Astronautics/International Society for Structural and Multidisciplinary Optimization (AIAA/ISSMO) Multidisciplinary Analysis and Optimization Conference*, MAO-10, Sep. 2008.
- [119] T. Zwinger, R. Greve, O. Gagliardini, T. Shiraiwa, and M. Lyly. A full stokes-flow thermo-mechanical model for firn and ice applied to the gorshkov crater glacier, kamchatka. *Annals of Glaciology*, 45:29–37, 2007.

Personal Facts

Nationality: Swiss

Date of birth: 20.02.1984

Native to: Villarod FR

Marital status: Fiancé

Education

EPFL

PhD in Mathematics, Numerical Analysis applied to Glaciology

Lausanne

2009–2013

The University of Fribourg

Master of Science in Physics, Minor of Mathematics

Trilingual studies in French, German, and English

Fribourg

2006–2008

The University of Fribourg

Bachelor of Science in Physics, Minor of Mathematics

Bilingual studies in French and German

Fribourg

2003–2006

Collège du Sud

Federal Maturity, Options Latin, Biology, Chemistry, and Physics

Bulle

1999–2003

Conferences

2013: SIAM Conference on Mathematical & Computational Issues in the Geosciences, Padova, Italy

Experience

Vocational.....

EPFL

Teaching Assistant

Teaching assistant in Analysis I, II, III, and Numerical Analysis for physicists, mathematicians, civil, mechanical, and environmental engineers.

Reference: Marco Picasso, EPFL, +41 21 693 42 47

Lausanne

2009–Present

The University of Queensland

Scientific Collaborator

Scientific collaboration with a research group at the University of Queensland where we worked on the design of efficient algorithms to simulate infinite, one-dimensional spin chains in the framework of Matrix Product States and Operators. Led to the paper in arxiv.org mentioned above.

Brisbane

2008–2009

Miscellaneous.....

SosCourrier Sàrl

Webmaster

Elaborated designs and interfaces of the web page of several societies.

Reference: Gilbert Michel, +41 26 652 22 38

Romont

2006–Present

Computer skills

Basic: OpenMP, CUDA, Mathematica, JavaScript

Intermediate: MPI, Python, Shell, Matlab, Octave, Linux OS

Advanced: C, C++, HTML, Php, MySQL, \LaTeX

Languages

French: Mothertongue

English: Intermediate

TOEFL iBT 95/120 in 2008

German: Intermediate

Scholar knowledge

Professional Interests

- Computer simulations of physical systems and their applications in the industry
- PDE-constrained optimization
- Parallel implementations of numerical algorithms
- Meteorology and climatology
- Data analysis
- Teaching